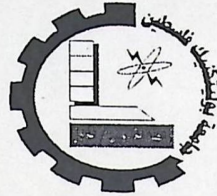


Palestine Polytechnic University



College of Engineering & Technology
Electrical & Computer Engineering

Graduation Project

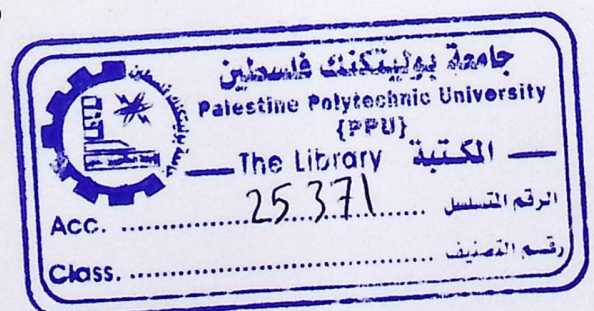
A Personal Computer Based General Purpose Control System
Using General Remote Control

Project Team

Ali M. Salameh
Omar G. Al-Rjoup
Hussein R. Al-Manasra

Project Supervisor
Eng. Mazan K. Zalloum

Hebron – Palestine
May, 2006



Abstract

The idea behind this project is using remote control works with infrared light, so it sent infrared light to receiver circuit connected with computer through serial port.

The user will be able to control PC remotely such as displaying start menu, open and close programs...etc.

On the other hand user able to control with application that connected with computer through parallel port such as home automation application.

We use VB.net to build the program that reads the digital signal (that's process by IRC) from serial port and process it to perform the related action , if the action is SW action the program call windows (.dll) file from system32 to perform this action ,if the action is HW action the program sent signal to the related pin in the parallel port, and the interface circuit amplify and process the signal to achieve the application result, and there is a heat sensor to stay a constant temperature level, so when the temperature increase the program terminate all hardware application.

تكمّن فكرة هذا المشروع في استخدام ريموت التحكم الخاص بالأجهزة الكهربائية المنزلية التي تعمل بالأشعة تحت الحمراء ، حيث يرسل أشعة إلى وحدة الاستقبال (IRC) متصلة بجهاز الكمبيوتر من خلال المنفذ المتسلسل ، التي تقوم بإدخال الإشارة إلى جهاز الكمبيوتر الذي يقوم بدوره بتحليل هذه الإشارة لتحديد الفعل (action) المرتبط بها ليتم تطبيقه على جهاز الكمبيوتر كعرض قائمة أبدا أو فتح برنامج معين ومن ثم إغلاقه ، وكذلك الأمر التحكم بالتطبيقات المتصلة بمنافذ الحاسوب.

ويتم ذلك من خلال برمجة برنامج باستخدام (VB.net) الذي يقرأ الإشارة من المنفذ المتسلسل على شكل إشارة رقمية التي تم تحليلها ومعالجتها باستخدام دائرة الاستقبال (IRC) ، ثم يقوم البرنامج بتحليل الإشارة لتنفيذ الفعل المرتبط بها ، فإذا كان الفعل (SW action) يقوم البرنامج باستدعاء برامج التحكم الخاصة لنظام التشغيل الموجودة على شكل ملفات (.dll) في المجلد (system32) ، أما إذا كان الفعل متعلق بالتطبيق المقترن بالمنفذ المتوازي (HW action) يقوم البرنامج بإخراج إشارة كهربائية إلى إحدى (Pins) الموجودة في المنفذ المتوازي ، وتقوم الدائرة الوسيطة (Interfacing circuit) بتكبير الإشارة ومعالجتها حسب التطبيق المطلوب ، وبهذا يتم التحكم بأي جهاز أو تطبيق مقترن بجهاز الكمبيوتر. كما ويوجد أيضا مجس حراري مرتبط بالمنفذ المتوازي ويتم قراءة الإشارة الناتجة منه باستخدام البرنامج ومن خلال هذه الإشارة يقوم البرنامج بالمحافظة على التطبيقات المرتبطة من خلال إيقافها إذا ارتفعت درجة الحرارة.

Contents

Title Page.....	I
Signature Page.....	II
Abstract.....	III
Dedication	V
Acknowledgments	VI
Table of Contents	VII
List of Figures	X
List of Tables.....	XI

1

Introduction

1.1 General Idea about Project and its Importance:.....	2
1.2 literatures Review:	2
1.3 System Requirements:	3
1.3.1 External Requirement:	3
1.3.2 Hardware Requirement:	4
1.3.3 Software Requirement:.....	4
1.3.3 Environment Requirements:.....	5
1.3.4 Communication Requirements:	5
1.4 Group Dependency:	5
1.5 Project Scheduling:	6
1.6 Project Cost:	10
1.7 Project Risks:.....	11
1.8 Report road map:.....	12

2

Theoretical Background

2.1 Introduction.....	15
2.2 Project Component.....	15
2.2.1 Transmitter Unit (Remote Control)	16
2.2.2 IR Receiver Unit.....	32
2.2.3 Control Unit	33
2.2.4 Application Unit.....	33

3 *Design Concepts*

3.1 Introduction.....	36
3.2 Objectives	36
3.3 General Block Diagram	37
3.3 How dose System Work?.....	39

4 *Hardware System Design*

4.1 Introduction.....	41
4.2 The units design	41
4.2.1 Remote control unit.....	42
4.2.2 IR Receiver unit:	44
4.2.3 Control unit.....	51
4.2.3 Application unit	61
4.3 Overall System Design.....	70

5 *Software System Design*

5.1 Introduction.....	73
5.2 General System Flowchart.....	73
5.3 System Operational Flowchart.....	74
5.3.1 Remote Control Initialization.....	75
5.3.2 Remote Signals Operations.....	77

6 *Implementation and Testing*

6.1 Introduction.....	80
6.2 Implementation.....	80
6.3 Testing.....	81

6.3.1 Testing the receiver circuit.....	81
6.3.2 Testing the optocouplers and relays.....	85
6.3.3 Testing the temperature circuit.....	86
6.3.4 Testing the program with the receiver circuit and application.....	87
6.3.5 Access the serial and parallel port using VB.net	87
6.4 Implementation and testing for Hardware component.....	88

7

Conclusion and Future Work

7.1 Introduction.....	91
7.2 Conclusions.....	91
7.3 Future view.....	92
Appendices.....	93
References	124

List of Figures

Figure 1.1: Project Activity Network	9
Figure 2.1: System Component	15
Figure 2.2: Electromagnetic Spectrum.....	23
Figure 2.3: The Electromagnetic Spectrum.....	25
Figure 2.4: Radio Frequency Energy Wave.....	26
Figure 2.5: ON / OFF Pulsing of IR light	27
Figure 2.6: IR signal carrier frequency	27
Figure 2.7: Encoder-Decoder set of IR remote control.....	29
Figure 2.8: Block Diagram of IR receiver.....	32
Figure 3.1: General Block Diagram.....	38
Figure 4.1: Remote Control Component	42
Figure 4.2: The Circuit Board	43
Figure 4.3: The Circuit Board	43
Figure 4.4: The Circuit Board	44
Figure 4.5: Infrared Receiver Circuit.....	45
Figure 4.6: TSOP Pins	46
Figure 4.7: TSOP Block Diagram	47
Figure 4.8 Single voltage regulator	49
Figure 4.9: Single Diode.....	49
Figure 4.10: Single capacitor	50
Figure 4.11: Resistors	50
Figure 4.12: Single Female connectors.....	51
Figure 4.13: Parallel Port Register	53
Figure 4.14: Serial port cable	57
Figure 4.15: Serial port pins.....	59
Figure 4.16: PC Com Port	60
Figure 4.17: Relay	62
Figure 4.18: Relay Parts.....	63
Figure 4.19: Optocouplers.....	65
Figure 4.20: AC motor circuit.....	67
Figure 4.21: Lamp circuit.....	68
Figure 4.22: Lamp circuit.....	69
Figure 4.23 Heat Sensor	69
Figure 4.24: The System Design Circuit.....	71
Figure 5.1 General System Flowchart.....	74
Figure 5.2 Remote Control Initialization Flowchart	75
Figure 5.3 Remote Signal Operation Flowchart	77
Figure 6.1: Infrared Receiver circuit	81
Figure 6.2: CMD screen.....	84

List of Tables

Table 1.1: The Task Duration.....	8
Table 1.2: project activity bar chart (First semester).....	9
Table 1.3: project activity bar chart (Second semester).....	10
Table 4.1: parallel port signal line	54
Table 4.2: register addresses of LPT1 and LPT2	56
Table 4.3:Serial Port Pins	61

1.2.	Literature Review
1.3.	System Requirements
1.4.	Group Dependency
1.5.	Project scheduling
1.6.	Project Cost
1.7.	Project Risk
1.8.	Report Road map

Introduction

1.1 General Idea about Project and its Importance:

Today, the remote control for electronic devices has become very ordinary. The widespread use of remote control units for TVs, VCRs, has been followed by its

1.1. General Idea about Project and it's Importance

1.2. Literature Review

1.3. System Requirements

1.4. Group Dependency

1.5. Project scheduling

1.6. Project Cost

1.7. Project Risk

1.8. Report Road map

At the end of this project the user can control with any application HW or SW using remote control, all of society levels can use and benefit from this project, it is services all of people by its applications, for example, the Handicap Can open door, put on light, etc ... from his chair

1.2 literatures Review:

There are some research people who have tried to use remote control to control of PCs and their applications. In palestiane polytechnic university they are working to do PC's Remote Control system by Asma' Abu Salah & Mahmood Ewawee ,in this project the user will be able to control the PCs remotely. He can perform many operations inside these PCs such as key press simulation, Mouse movement...etc. And the Microcontroller is the main control unit that handles the system. It receives the data from the remote control and transmits it to one of the PCs.

1.3 System Requirements:

The system requirements of this project include several parts each of them related together to form the whole system in order to classify these requirements are:

1.3.1 External Requirement:

Our project needs some requirements to run it , such of these:

1. Remote control to select the action you wanted to do.
2. Visual basic .net to develop the interface program.
3. The infrared receiver circuit resides to the front of the single pc and in the direction of the remote to receive the signal from the remote.
4. The user has to setup the system before starting using and this require several activities some of these activities:

- Check remote state (battery mater, workable, not workable)
- Run the computer to check workability.
- Run the screen and adjust colors, resolutions.
- Run the system and start viewing.

1.3.2 Hardware Requirement:

These are the requirement are required by the final project to run on the user computer to use and run such of these requirements are:

1. PC Pentium (III) 1000 MHZ speed 256kb cash memory or up.
2. 15 inch screen or larger.
3. 128 MB SDRAM or higher.
4. 20 GB hard disk or higher.

1.3.3 Software Requirement:

After analyzing all software needs and requirements, the following software functions and modules are needed:

- **Win ME/NT/2000/XP:**

This system needs one of these types of windows to use it as environment to run our system under it.

- **Microsoft Visual Studio.NET:**

We need to use this compiler to build and run our software driver which programmed using VB.NET

1.3.3 Environment Requirements:

The environment requirements include the region where the system to be installed not more than 30m between the remote control and receiver circuit. Another aspect around the system environment to be no persons or parts between the remote and the infrared receiver circuit other wise the signal may not be sent correctly.

1.3.4 Communication Requirements:

The communication requirements included in infrared technology; so there is no need for any cabling or media device the system to communicate with its part correctly there must be straight line between remote and infrared receiver circuit to send and receive signal correctly.

Also to prevent other noisy resources that may cause signal damage or destroying since the system send and receive through the air, so that the distance between the remote and infrared receiver should be with few meters.

1.4 Group Dependency:

The activity of the project has been distributed to the group as separate parts to be designed separately and after completing each part to be tested as follows:

- Infrared Receiver Circuit is accomplished by Ali Salameh.
- Software Driver (SD) is accomplished by Omar Al Rujob.
- Application is accomplished by Hussein Manasrh.

After these parts have been accomplished by the partners, the group will meet to integrate these parts.

1.5 Project Scheduling:

The timing management will divide the system hierarchy according to the dependent and nondependent actions, this process will explain the critical path of the system.

The project activities here depend on each other, so the task durations and dependencies are the following:-

T1: Preparing to the project: here we introduce to start in the project and discuss with the advisor to initialize the project, and preparing the group and evaluate the project tasks and levels. At this period we explain the system requirements.

The project consists of the four major high level parts: the infrared Receiver circuit, Infrared transmitter Circuit (the Remote circuit), the Software Driver and Applications.

T2: The project searching and analysis: at this period we start the first step to search and analysis the project and allocate information and data about the project levels

and sublevels, tasks and subtasks, there are many resources to searching and analyzing the concepts.

T3: the project requirements analysis: the project has many tasks and equipments must be provided and explained to implement the final project to achieve the system requirements and tasks. The system has a hardware and software requirements which must be achieved through the prototype and final presentation.

T4: Infrared receiver Circuit: This circuit will receive the signal sent from the remote side.

T5: Remote control: we need IR remote control to work as transmitter.

T6: Software Driver: software program that can be setup on the PC which allows the acquired signal from the infrared receiver circuit to be performed as computer function.

T7: Application: here we need to construct and build the circuits that represent the home automation system.

T8: The prototype: in the prototype we will display to the user, the project efficiency and his requirements that will implemented correctly in the project, so this step is mainly because it will reflect the user opinion and the project manipulation.

T9: Test the project: we will test the project and implement it to insure that the system and user requirements levels are achieved or not, to adjust the problems and errors in the project and can maintain it, then try to test and execute it again

T10: Re_analyse and re-implement the project

T11: Final Project and presentation: the final project will be display completely without any problem and achieve the highest level objectives, the user can press on any remote key and the receiver circuit will receive the data and translate it to the computer which will control the Application.

T12: Documentation writing: the writing begins with us from the first step to the last one in parallel.

Table 1.1: The Task Duration

Task	Duration(days)	Dependencies
T1	15	----
T2	15	-----
T3	20	T1,T2
T4	35	T3
T5	10	T4
T6	45	T4
T7	40	T5
T8	15	T4,T5
T9	15	T8
T10	5	T9
T11	10	T10
T12	150	T11

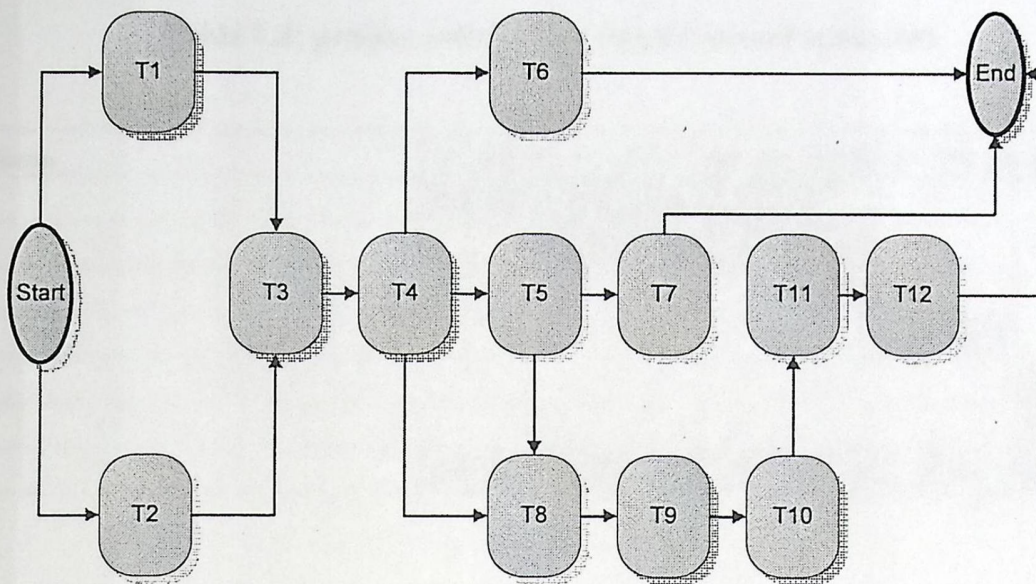


Figure 1.1: project activity network

Table 1.2: project activity bar chart (First semester)

Task / Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
T1	■	■	■													
T2				■	■	■										
T3					■	■	■	■								
T4									■	■	■	■	■	■		
T5														■	■	
T12	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	

Table 1.3: project activity bar chart (Second semester)

Task / Week	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
T6																
T7																
T8																
T9																
T10																
T11																
T12																

1.6 Project Cost:

We purchase as a work group all of its equipments to complete the project architecting and designing.

The project need both of hardware equipments and software programs that runs on the computer, so we will purchasing all needed electronic components and parts and software programs.

- The Hardware Components, there are many electrical Chips and equipments have to be provided Such as:-

1. Resistors.
2. Capacitors.
3. Diodes.
4. Infrared TSOP 173 (5\$).
5. TV Remote (5\$).
6. Computer Pentium two or more (500\$).
7. Printer Cable (5\$).

8. Breadboard (3\$).
9. Wires (3\$)
10. (8) Optocouplers (5\$)
11. (3) Relays (25\$).
12. Fan (15\$)
13. DC Motor (100\$)
14. (2) lamps (3\$)
15. Triac (2\$)
16. Heat sensor (5\$)

▪ Software programs Such as:-

1. Visual Basic.Net Program it cost (700\$).
2. Windows 98version or more it cost (500\$).
3. Microsoft PowerPoint, word, visio, and publisher (300\$).

The total cost: - the total cost contains the hardware equipments and software programs, the total cost reach about approximately (2000\$)

Note: the Electronic equipments price is varied depending on the component Efficiency and the purchasing source and as the performance of the project increase the cost increase.

1.7 Project Risks:

The project must avoid some of problems to work as its high efficiency, so there are some of these risks may be occur and we have to declare them in the early time of the project designing and manipulation, when we find a risk we will try to solve it without effect on the total project as we can.

1. The range of the transmitter waves may be less than the actual distance between the computer and the remote.
2. The receiver circuit do not response, or may be don't understand or agree with the coming waves, so it will make a problem.
3. The needed time for response is so long that makes the user nervous.
4. The scheduling and dependency of the all system parts must be in the correct direction and expectation, otherwise we will fault and late in producing the complete and integrated program. like the receiver circuit depends on the circuit interface, if a problem is occur to the circuit interface it refers to no executed system.
5. Some of project parts do not work correctly; it will break the whole system, E.g. if the capacitor does not wok or disconnect in the receiver circuit.

1.8 Report road map:

Report consists of seven chapters; the following is a brief description of the topics that are covered in each chapter.

Chapter 1: Introduction

This chapter present general idea about the project and its importance, and also literature review, system requirement, group dependency, project scheduling, estimated cost.

Chapter 2: Theoretical Background

This chapter talks in more details about the basic component used in the project and theoretical back ground.

Chapter 3: Design concepts

This chapter details the design concepts, introduces project objectives, shows the general block diagram of the system and explains how system works.

Chapter 4: Hardware System Design

This chapter presented out lines formal procedure for design, discuss design options and justify those chosen for the project.

Chapter 5: Software System Design

This chapter handles the software related to our system, depicts flowcharts about system operation.

Chapter 6: System Implementation and Testing

This chapter includes the implementation phase with the testing of these phase. General hardware and software component tested and shown in this chapter.

Chapter 7: Conclusion and Future Work

This chapter will provides the conclusions that will be concluded after working the system, and suggestion for future work.

Theoretical Background

2.1 Introduction.

2.2 Project Component.



Figure 2.1: System Component

Chapter Two

The Theoretical Background

2.1 Introduction

This Chapter focuses on the theoretical subject related to the main idea of the project, and information about the components used in the project.

2.2 Project Component

This project has many components which are listed in the following section as shown in the figure 2.1, this components are:

- Transmitter Unit (Remote Control).
- IR Receiver Unit.
- Control Unit (PC).
- Application Unit (Home Automation).

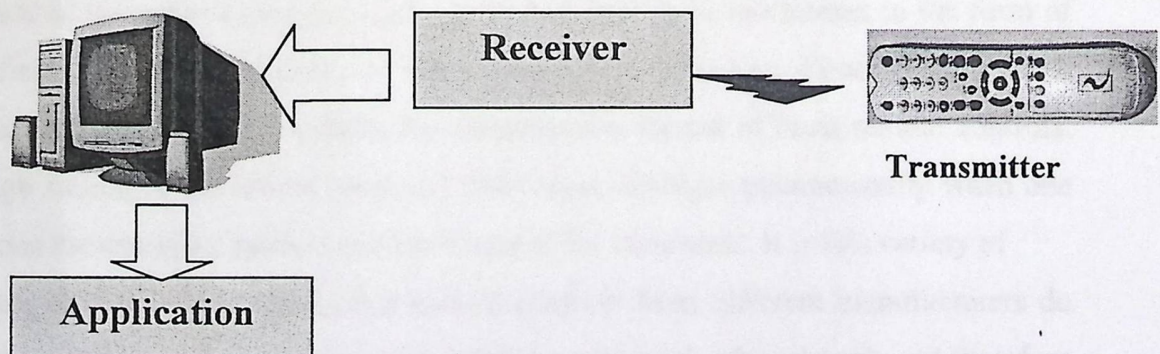


Figure 2.1: System Component

2.2.1 Transmitter Unit (Remote Control)

Historically consumer appliance remote controls were developed to provide a convenient way to control the appliance without having to move near it. Early remote controls were essentially detachable control panels connected to the appliance through a thick umbilical cord.

The introduction of solid state electronics and integrated circuits allowed the development of battery-powered wireless remote controls. Although, radio frequency and ultrasound technologies have been used in the past to provide the communication path between the control and the appliance, nowadays virtually all appliance remote controls are based on optoelectronic components communicating using infrared light signals.

The evolution of appliance controls currently moves towards the transfer of functionality from the appliance to the remote control. A number of modern appliances provide only minimal or no control on the main appliance and delegate all functionality to the remote control. Notable such examples are VCRs, televisions, and air-conditioners. Another interesting and helpful for our application trend is the inclusion of remote control interfaces on appliances where such a control might seem illogical such as car radios and personal stereos. Most remote controls provide essentially one-directional links between the control and the appliance. The reverse feedback of the control loop is usually provided directly to the human in the form of an indicator on the appliance, or altered appliance behavior. Contrary to what a rational technologist might expect, the transmission format of most remote controls, although similar at the lowest (physical link) layer, diverges phenomenally when one examines the encoding method used to transmit the command. It is this variety of encoding methods that ensures that remote controls from different manufacturers do not control other appliances (they still interfere with each other though and therefore

can not be used at the same time). The majority of remote controls encode commands by pulse modulating a 980nm infrared light signal at a carrier frequency ranging from 32 KHz to 40 KHz.

The modulated sequence typically transmits a binary word using a self-clocking mechanism. Word length varies from 12 to 48 bits. At the beginning of each word a continuous signal burst is often used as a mechanism to tune the receiver's automatic gain control circuit. Bits are encoded using either pulse width or diphase Encoding. More complicated schemes such as clock-based serial transmission are often used for communication between more intelligent devices such as computers with infrared ports.

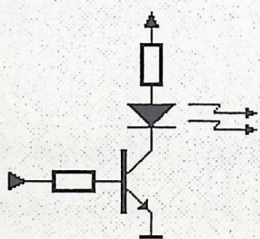
2.2.1.1 How does transmitter works?

The transmitter usually is a battery powered handset. It should consume as little power as possible, and the IR signal should also be as strong as possible to achieve an acceptable control distance. Preferably it should be shock proof as well.

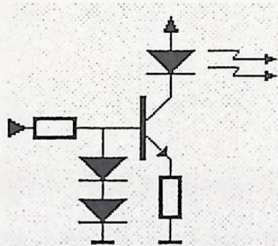
Many chips are designed to be used as IR transmitters. The older chips were dedicated to only one of the many protocols that were invented. Nowadays very low power microcontrollers are used in IR transmitters for the simple reason that they are more flexible in their use. When no button is pressed they are in a very low power sleep mode, in which hardly any current is consumed. The processor wakes up to transmit the appropriate IR command only when a key is pressed.

Quartz crystals are seldom used in such handsets. They are very fragile and tend to break easily when the handset is dropped. Ceramic resonators are much more suitable here, because they can withstand larger physical shocks. The fact that they are a little less accurate is not important.

The current through the LED (or LEDs) can vary from 100mA to well over 1A! In order to get an acceptable control distance the LED currents have to be as high as possible. A trade-off should be made between LED parameters, battery lifetime and maximum control distance. LED currents can be that high because the pulses driving the LEDs are very short. Average power dissipation of the LED should not exceed the maximum value though. You should also see to it that the maximum peak current for the LED is not exceeded. All these parameters can be found in the LED's data sheet.



A simple transistor circuit can be used to drive the LED. A transistor with a suitable HFE and switching speed should be selected for this purpose. The resistor values can simply be calculated using Ohm's law. Remember that the nominal voltage drop over an IR LED is approximately 1.1V.



The normal driver, described above, has one disadvantage. As the battery voltage drops, the current through the LED will decrease as well. This will result in a shorter control distance that can be covered.

An emitter follower circuit can avoid this. The 2 diodes in series will limit the pulses on the base of the transistor to 1.2V. The base-emitter voltage of the transistor subtracts 0.6V from that, resulting in constant amplitude of 0.6V at the emitter. This constant amplitude across a constant resistor results in current pulses of a constant

magnitude. Calculating the current through the LED is simply applying Ohm's law again.

2.2.1.2 Type of Transmission Signals

There are many types for signal transmission these types are:

□ Cables:

A wired connection, like optical fiber, provides a virtually noise free control channel. Also, since no transducers are involved, the circuitry is simpler and there is usually no need for amplification. The cables may pick up noise in the form of electromagnetic interface, but using shielded cables can minimize the noise. Also using the cables are useful for controlling fixed devices that have several functions to be controlled.

□ Ultrasound:

Ultrasound is sound of such high frequency, that is, it can not be heard by the human ear. Atypical ultrasound transducer operates at a frequency of 40 KHz but the ear can hear up to 20KHz. Ultrasound transducer is relatively inexpensive and the circuit required is simple to build.

□ **Radio Frequency (RF):**

Radio control has the advantage of the operating over a far greater range than ultrasound, also it does not required the physical connection as in the cable control, it is very popular for the control of aircraft's, boat's.....etc.

□ **RF Advantages:**

- Not line of sight.
- Not blocked by common materials: can penetrate most solids and pass through walls.
- Longer range.
- Not light sensitive.
- Not as sensitive to weather/environmental conditions.

□ **RF Disadvantages:**

- Interference: communication devices using similar frequencies - wireless phones, scanners, wrist radios and personal locators can interfere with transmission.
- Lack of security: easier to "eavesdrop" on transmissions since signals are spread out in space rather than confined to a wire.
- Higher cost than infrared.
- Federal Communications Commission (FCC) licenses required for some products.
- Lower speed: data rate transmission is lower than wired and infrared transmission.

❑ **Bluetooth Advantages:**

- Its wireless. Eliminate messy and confusing cables at home, the office, or when traveling.
- It's inexpensive. Vendors can incorporate it into their products and consumers can upgrade older equipment cheaply.
- Uses radio signals so can pass through walls and does not require line of sight.
- Lower power consumption. Won't drain your battery.
- 2.5 GHz radio frequency ensures worldwide operation.
- No thinking required. The devices find one another and connect without any user input at all.

❑ **Bluetooth Disadvantages:**

- New type of technology and must be accepted unequivocally by all vendors and manufacturers in order to guarantee compatibility among the array of products.
- Designed for only short-range communications certified to no more than 100 meters (with hub broadcaster.)
- Data transfer speeds are not as fast as other wireless technologies.

❑ **Infrared (IR):**

With the aid of a glass prism we can demonstrate that the white light coming from the sun is actually made up of many different colors some of the light falls into the visible portion of the spectrum while wavelengths, such as the infrared and

ultraviolet rays, remain invisible. The distribution of electromagnetic spectrum is shown in figure-2.2 the spectrum that lies just outside the human eye red sensitivity limit is called “near infrared” or simply IR. It is this portion of the spectrum that is used by much of today’s light-beam communications systems. As can be seen from figure-2.2, sunlight is a very powerful source for this band of light, so are standard incandescent lamps and light from camera photo-flash sources. However, many other man-made light emitters, such as fluorescent lamps and the yellow or blue/white street lamps, emit very little infrared light. But the question is why we cannot see infrared light? Referring to figure-2.2, which shows the human eye response to the light, we see that there, is no relative.

Sensitivity of the human eye to the range of infrared light. So, to detect infrared light, we must use a special detector that have a maximum sensitivity in the infrared region.

Figure 2.2: electromagnetic spectrum

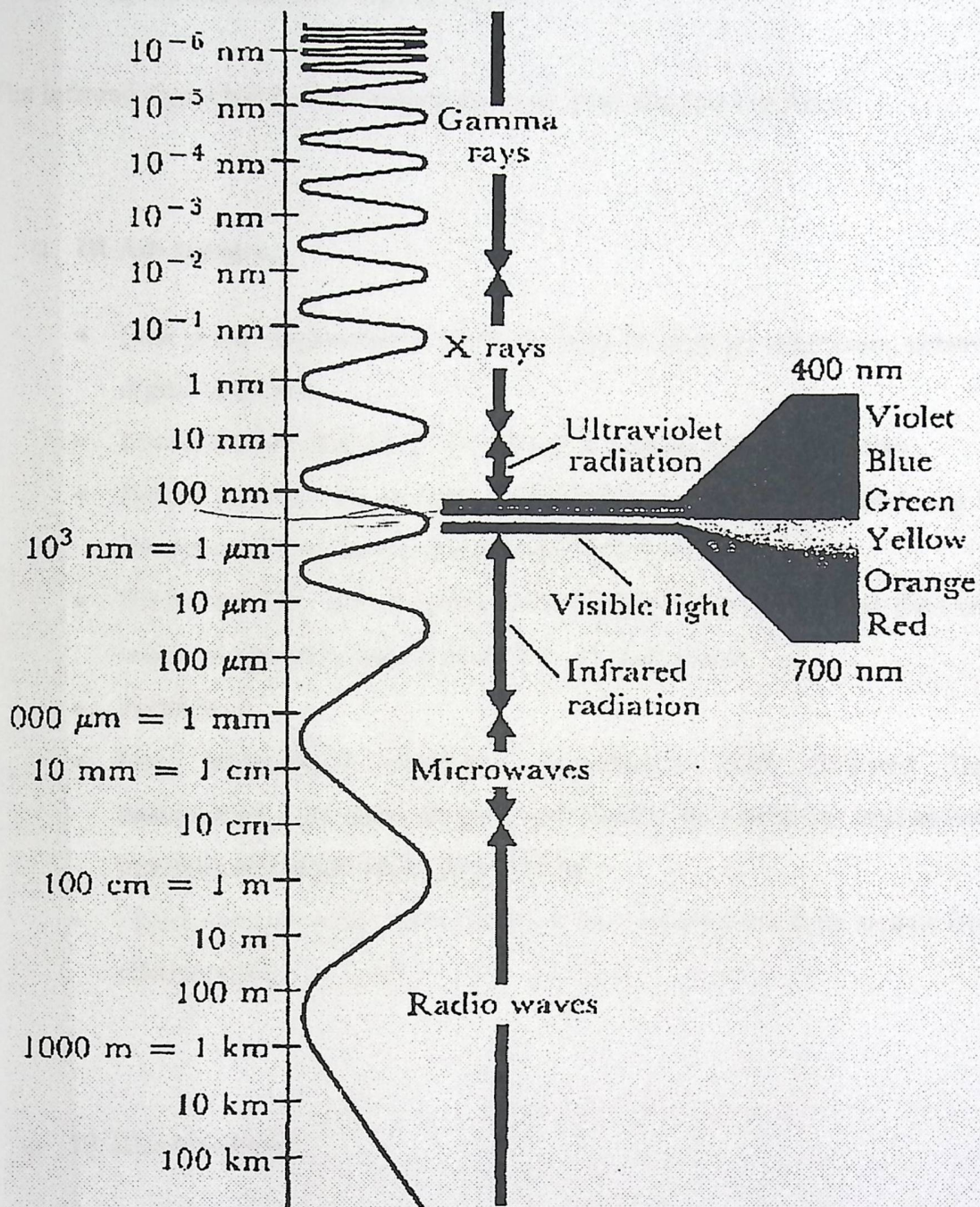


Figure 2.2: electromagnetic spectrum

□ Why we use infrared transmission?

The infrared signal has a lot of advantages over other one some of these:

□ IR Advantages:

- Low power requirements: therefore ideal for laptops, telephones, personal digital assistants.
- Low circuitry costs: \$2-\$5 for the entire coding/decoding circuitry
- Simple circuitry: no special or proprietary hardware is required, can be incorporated into the integrated circuit of a product.
- Higher security: directionality of the beam helps ensure that data isn't leaked or spilled to nearby devices as it's transmitted.
- Portable.
- Few international regulatory constraints: IrDA (Infrared Data Association) functional devices will ideally be usable by international travelers, no matter where they may be.
- High noise immunity: not as likely to have interference from signals from other devices.

□ IR Disadvantages:

- Line of sight: transmitters and receivers must be almost directly aligned (i.e. able to see each other) to communicate
- Blocked by common materials: people, walls, plants, etc. can block transmission
- Short range: performance drops off with longer distances

- Light, weather sensitive: direct sunlight, rain, fog, dust, pollution can affect transmission
- Speed: data rate transmission is lower than typical wired transmission

□ Comparison between IR and Bluetooth and RF:

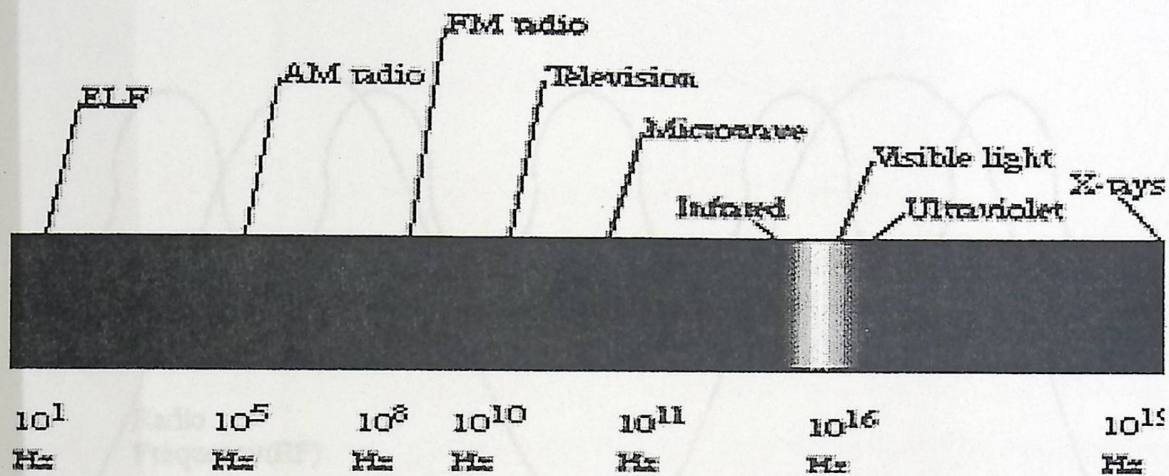


Figure 2.3: The electromagnetic spectrum

The electromagnetic spectrum is depicted in Figure 2.3. A horizontal bar represents a range of frequencies from 10 Hertz (cycles per second) to 10 to the 18th power Hertz. Some familiar allocated frequency bands are labeled on the spectrum. Approximate locations are as follows. (Exponential powers of 10 are abbreviated as 10exp.)

10 Hertz: extremely low frequency or ELF.

10exp5 Hertz: AM radio.

10exp8 Hertz: FM radio.

10exp10 Hertz: Television.

10exp11 Hertz: Microwave.

10exp16 Hertz: Infrared (frequency range is below the visible light spectrum).

10exp16 Hertz: Visible Light.

10exp16 Hertz: Ultraviolet (frequency range is above the visible light spectrum).

10exp18 Hertz: X-rays.

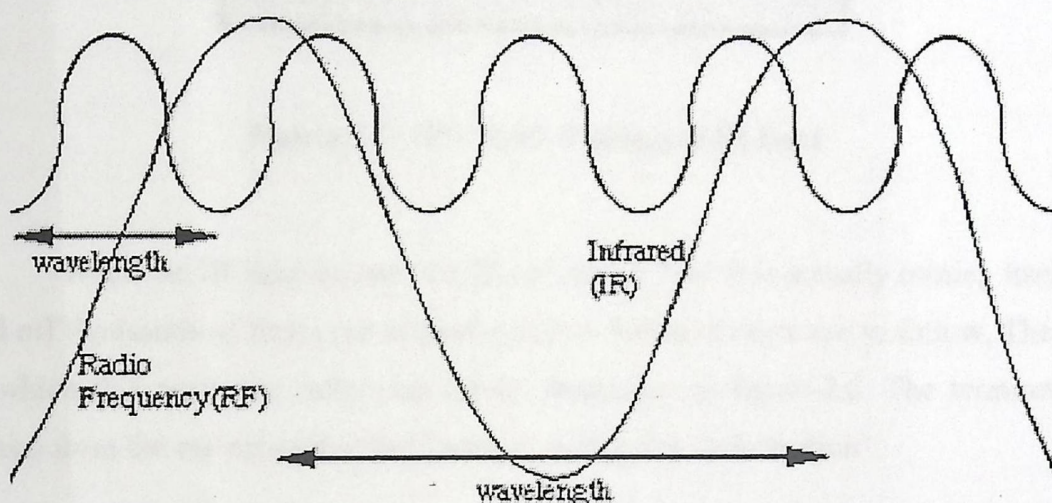


Figure 2.4: radio frequency energy wave

Figure 2.4 depicts a radio frequency energy wave superimposed upon an infrared energy wave, and illustrates the inverse relationship between frequency and wavelength. The infrared energy wave completes nearly 5 and a half cycles in the time that the radio frequency wave completes 2 cycles. The wavelengths of the infrared wave and the radio wave are labeled, and the infrared wavelength is less than half the wavelength of the radio wave.

□ **Infrared Signaling and how it works:**

IR remotes operate by modulating (turning on and off) an infra red (IR) light source as figure-2.5.

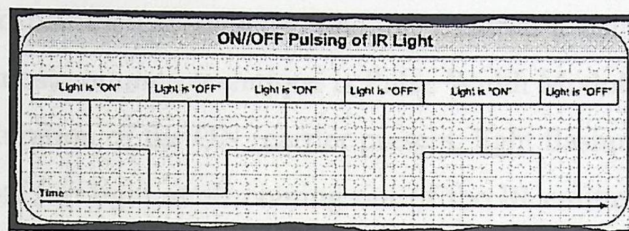


Figure 2.5: ON / OFF Pulsing of IR light

When the IR light source (the IR emitter) is "on" it is actually turning itself on and off thousands of times per second, too fast for the human eye to follow. The rate at which this occurs is called the carrier frequency as figure-2.6. The terminology comes from the metaphor that the "carrier" carries the "information".

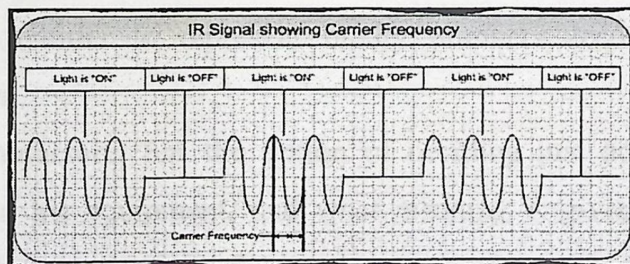


Figure 2.6: IR signal carrier frequency

This is done to provide a better transmission system and allow the overall IR system (transmitter and receiver) to operate in noisy (with respect to light) environments. It is important to understand that the IR receiver for a given remote is tuned to IR "carrier" frequency for that remote and will effectively not see IR signals

sent on a different carrier frequency such as from other remotes. [Note: The human eye can never see an infrared transmission, so the concept of on and off is not with regards to visible light]. Some equipment has a "telltale", a little red light that visibly flashes when the equipment receives IR signals.

The "information" is placed on the "carrier" using several different techniques. The most common technique is Pulse Width Modulation. In Pulse Width Modulation the duration of the ON (carrier present, light flashing thousands of times per second), or off (no light at all coming out of the IR emitter) periods is made to vary. Let's assume, because this is what is done, that we wish to send numbers representing what key has been pressed (and perhaps even what device this key is for). We first need to simplify the problem so that we don't have deal with too many "Pulse widths". We can easily do this by representing the number in base 2, or binary. In binary there are only two digits to worry about not ten as in decimal. Therefore we only need to have two distinct "pulse widths". If you think about it, the periods of on and off will need to alternate. If they didn't it would be hard to judge their width.

In summary, IR transmission most often takes place by varying the on off times of an IR emitter to represent binary numbers according to some well established pattern.

Each manufacturer has the option of deciding just how big a number he wishes to send to his equipment, and what meaning is given to that number (or numbers) when they are received. Remember the environment through which the IR signals are passing (the air) is noisy in a light sense. Bright sunlight, Fluorescent lights, all contributes to the noise. Some manufacturers add additional "redundant" information such as sending the numbers twice to ensure that they get to the equipment correctly.

□ 1R Remote Control Decoding:

All Infrared (IR) remote controls use some kind of IR signal. The remotes transmit pulses of IR light to send the signal to the receiver. These IR LEDs transmit light in the frequency range of 30 kHz to 40 kHz. These high frequencies were chosen so that other light sources would not interfere with the receiver's ability to correctly receive the transmitted signals.

These signals are transmitted by the IR led in some type of binary code. It turns out that for most consumer electronics this coding is the same. The binary signal varies in length for both time and bit length. Figure 2.7 shows Encoder-Decoder set of IR remote control.

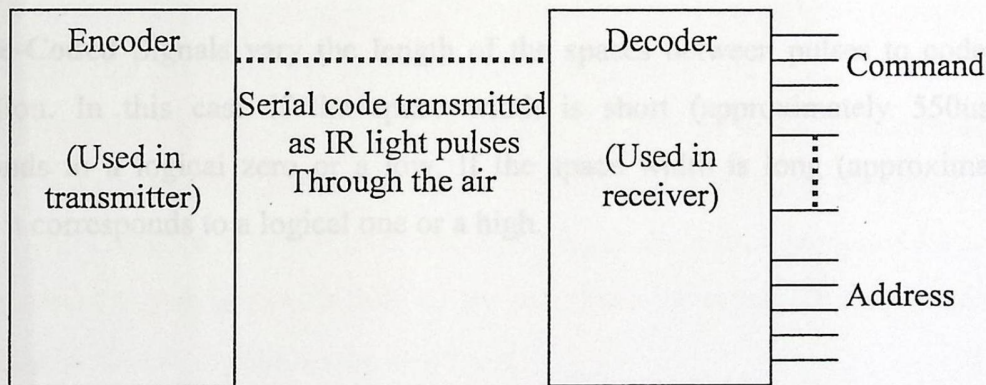
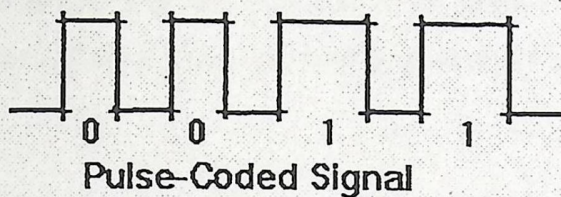


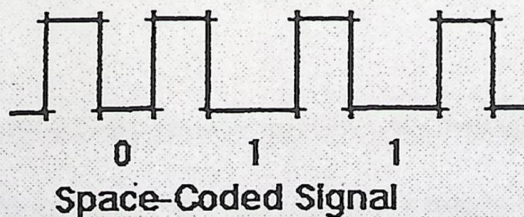
Figure 2.7: Encoder-Decoder set of IR remote control

The following is some methods to that are used to represent the binary information in decoding the remote controls.

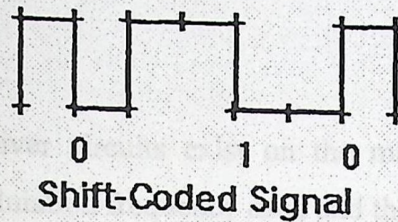
1) **Pulse-Width-Coded Signals** vary the length of pulses to code the information. In this case if the pulse width is short (approximately 550us) it corresponds to a logical zero or a low. If the pulse width is long (approximately 2200us) it corresponds to a logical one or a high.



2) **Space-Coded Signals** vary the length of the spaces between pulses to code the information. In this case if the space width is short (approximately 550us) it corresponds to a logical zero or a low. If the space width is long (approximately 1650us) it corresponds to a logical one or a high.



3) **Shift-Coded Signals** vary the order of pulse space to code the information. In this case if the space width is short (approximately 550us) and the pulse width is long (approximately 1100us) the signal corresponds to a logical one or a high. If the space is long and the pulse is short the signal corresponds to a logical zero or a low.



This coding holds information such as the address to the machine that is using the remote and the command that the machine must follow. The address is very important because without it the signal would be processed by another IR receiver in the area.

Figure 2.8: block diagram of IR receiver

When a button on a remote is pushed it sends a string of signals. The first piece of information in the string is called the Header. The Header usually contains a burst of highs that alerts all the IR receivers in the area to the string of data being sent. Following the burst of highs is the address to the specific machine to receive the next piece of data, the command. As long as the button is held down (depressed) the command will continue to repeat over and over. When the button is released a string of code called the stop is transmitted. As we may have expected, the stop code tells the machine to stop its executing the command.

So, we can conclude that the remote control code consists of three parts:

1. **A header:** to activate all the JR receivers in the area to the string of code being sent.
2. **An Address:** selects which equipment to send to.
3. **A Command:** tells the equipment what to do.

2.2.2 IR Receiver Unit

Many different receiver circuits exist on the market. The most important selection criteria are the modulation frequency used and the availability in your region.

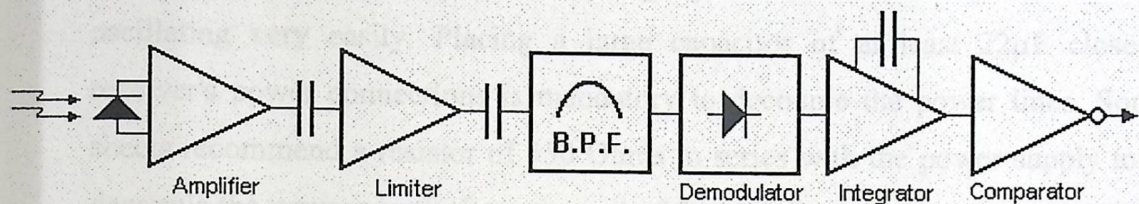


Figure 2.8: block diagram of IR receiver

In the Figure-2.8 above you can see a typical block diagram of such an IR receiver. Everything is built into one single electronic component.

The received IR signal is picked up by the IR detection diode on the left side of the diagram. This signal is amplified and limited by the first 2 stages. The limiter acts as an AGC circuit to get a constant pulse level, regardless of the distance to the handset.

As you can see only the AC signal is sent to the Band Pass Filter. The Band Pass Filter is tuned to the modulation frequency of the handset unit. Common frequencies range from 30 kHz to 60 kHz in consumer electronics.

The next stages are a detector, integrator and comparator. The purpose of these three blocks is to detect the presence of the modulation frequency. If this modulation frequency is present the output of the comparator will be pulled low.

As we illustrate before, all these blocks are integrated into a single electronic component. There are many different manufacturers of these components on the market. And most devices are available in several versions each of which are tuned to a particular modulation frequency.

The amplifier is set to a very high gain. Therefore the system tends to start oscillating very easily. Placing a large capacitor of at least $22\mu\text{F}$ close to the receiver's power connections is mandatory to decouple the power lines. Some data sheets recommend a resistor of 330 Ohms in series with the power supply to further decouple the power supply from the rest of the circuit.

2.2.3 Control Unit

The control unit represent by PC computer and interfacing circuits. The PC computer represents the main brain of the control unit on one hand, and on the other hand represents the interfacing circuits that integrated the system.(we will explain this unit in detail at chapters four and five).

2.2.4 Application Unit

The application unit represents smart home, that means controlling of objects through the computer, using the signals which is send from remote control's user be able to demonstrate the functionality of these objects.

And this make the life easier and it will offer less cost for large companies, so our project will be helpful for a large verity of society.

3

The controlling of the system will be automatically in some parts that's mean the PC will manage the job of the connected devices as a stand alone system, so no need for an expert to work with this system each time.

The project also will have a wide control of the components connected to it, through hardware circuits that compatible with software. (We will explain this unit in detail at chapter four).

3.1 Introduction

3.2 Project Objectives

3.3 General Block Diagram

3.4 How System Works

Design Concepts

- 3.1 Introduction
- 3.2 Project Objectives
- 3.3 General Block Diagram
- 3.4 How System Works

Chapter Three

Design Concepts

3.1 Introduction

In this chapter we are going to describe the main objectives of the system and the general block diagram.

3.2 Objectives

This project supports many ideas and objectives that can be summarized as follows:

- 1- The user can use remote technique to run several applications on computer directly with out need to use keyboard or mouse, instead of that we uses remote button to run several applications.
- 2- The user can use remote to control of any hardware application connected with computer ports such as open and close dooretc.
- 3- To make remote technology available in computer field so any one can use and run this technology with out difficulties and provide software need to run remote circuit under computer area.
- 4- To design general purpose remote control with lowest costs.
- 5- Make life easier.

3.3 General Block Diagram

The following figure-3.1 show block diagrams our system. It consists of several units to be accomplished and integrated to with each other to form the final remote control system these units are:

- Remote Control the device that to be used to send signal to the (IRC).
- Infrared receiver circuit (IRC) the circuit we went to construct it to receives the signal from remote and send it back to the computer through serial port.
- Software Driver the program that receives the signal from (IRC) and decode the signal then perform the specific task (SD).
- Application (Home automation).

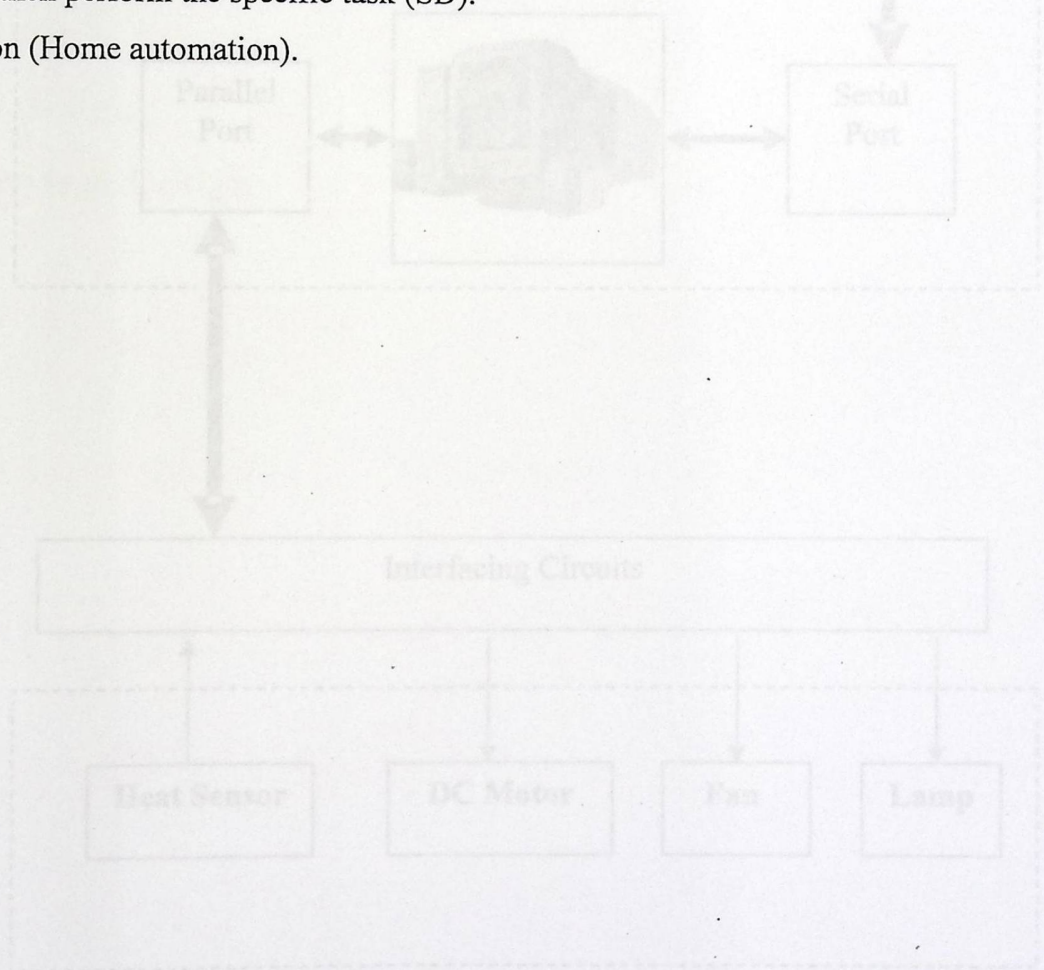


Figure 3.1: General Block Diagram

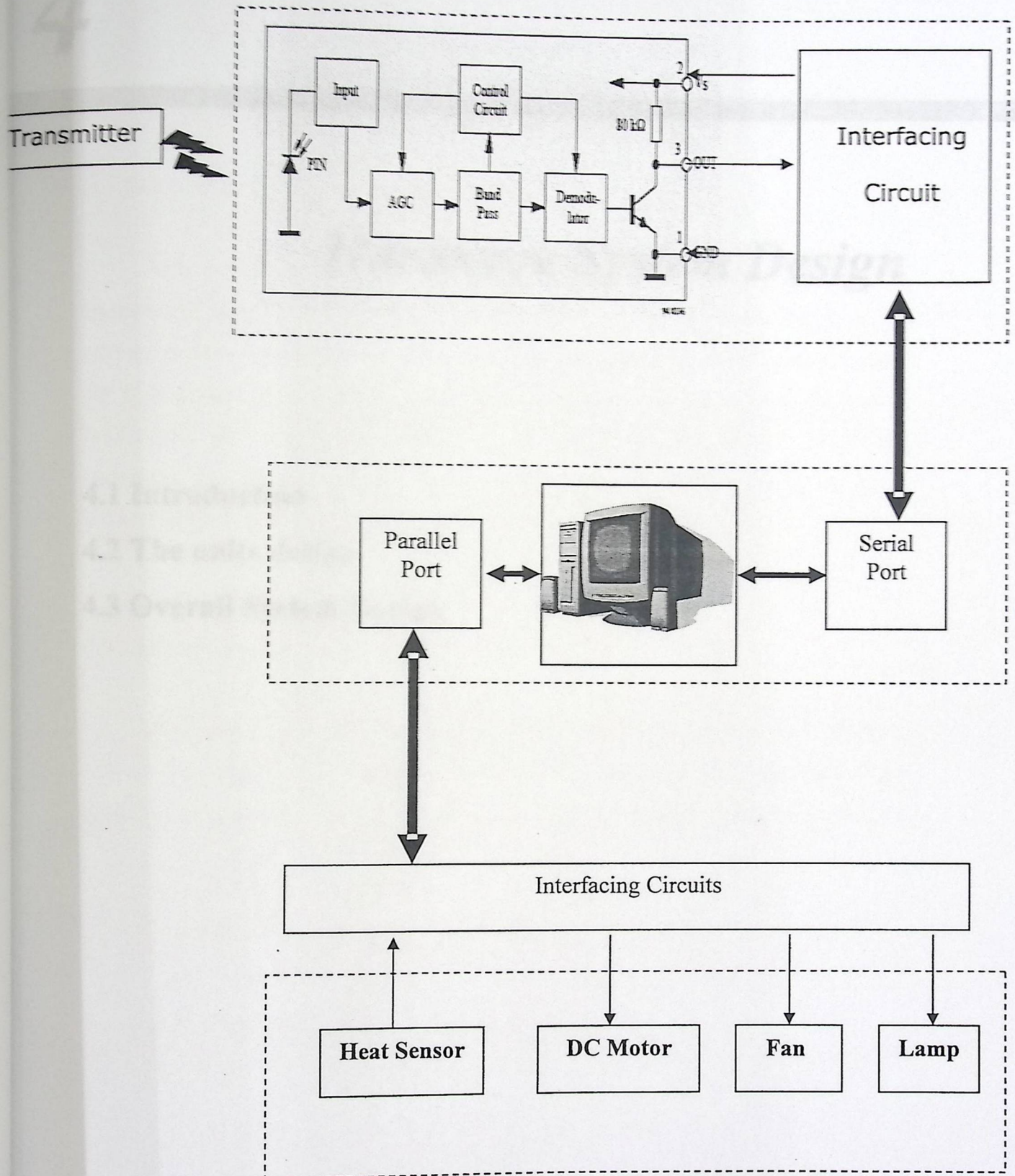


Figure 3.1: General Block Diagram

Hardware System Design

4.1 Introduction

4.2 The units design

4.3 Overall System Design

- Remote control unit.
- IR Receiver unit.
- Control unit.
- Application unit.

Chapter Four

Hardware System Design

4.1 Introduction

After explaining the theoretical background, the general block diagram of the system, and how the system works, there is a need to view what is the design of this system in more specific, powerful and more formal terms. So this chapter describes the final system design with all its features which are necessary to make the system works well and achieves the objectives of the system. This chapter shows the interfaces between the equipments of the theoretical background, and the more suitable chips that advanced the design.

4.2 The units design

The system has mainly four parts; these parts must interface with each others to achieve the project goals. The team has to deal with each one as separate unit, study its features and prepare it to operate successfully with other units. These units are:

- Remote control unit.
- IR Receiver unit.
- Control unit.
- Application unit.

4.2.1 Remote control unit

The remote control's job is to wait for you to press a key, and then to translate that key-press into infrared (pronounced "infra-red") light signals that are received by the IR Receiver. When you take off the back cover of the remote control you can see that there is really just one part visible: a printed circuit board that contains the electronics and the battery contacts.

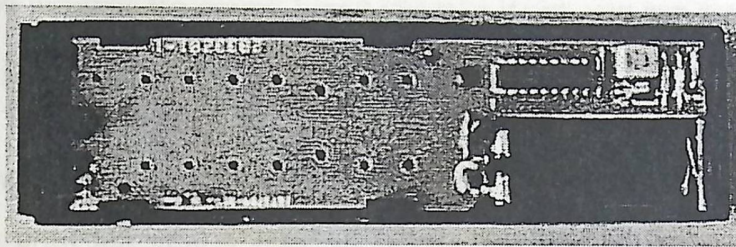


Figure 4.1: remote control component

The components that you see figure-4.1 are typical for most remotes. You can see an integrated circuit (also known as a chip). The chip is packaged in what is known as an 18 pin Dual Inline Package, or a DIP. To the right of the chip you can see a diode, a transistor (black, with three leads), a resonator (yellow), two resistors (green) and a capacitor (dark blue). Next to the battery contacts there is a resistor (green) and a capacitor (tan disk). In this circuit, the chip can detect when a key is pressed. It then translates the key into a sequence something like Morse code, with a different sequence for each different key. The chip sends that signal out to the transistor to amplify the signal and make it stronger.

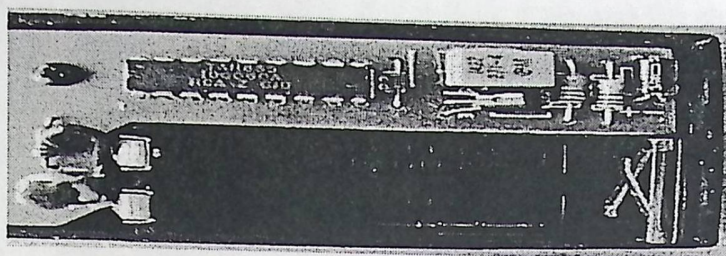


Figure 4.2: The Circuit Board

When you unscrew the circuit board and take it out, you can see that the circuit board is a thin piece of fiber glass that has thin copper "wires" etched onto its surface see figure 4.2. Electronic parts are assembled on printed circuit boards because they are easy to mass produce and assemble. In the same way that it is relatively inexpensive to print ink onto a sheet of paper, it is inexpensive to "print" copper wires onto a sheet of fiber glass. It is also easy to have a machine drop the parts (the chips, transistors, etc.) onto the sheet of fiberglass and then solder them on to connect them to the copper wires.

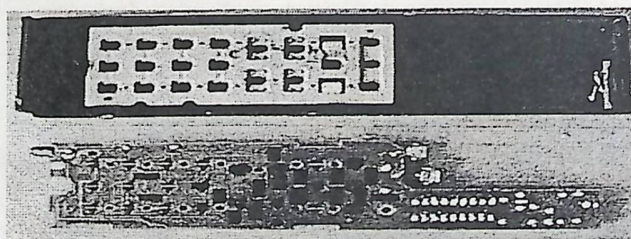


Figure 4.3: The Circuit Board

When you look at the board, you can see a set of contact points for the buttons as figure 4.3. The buttons themselves are made of a thin rubbery sheet. For each button there is a black conductive disk as shown in figure 4.4. When the disk touches

the contacts on the printed circuit board, it connects them and the chip can sense that connection.

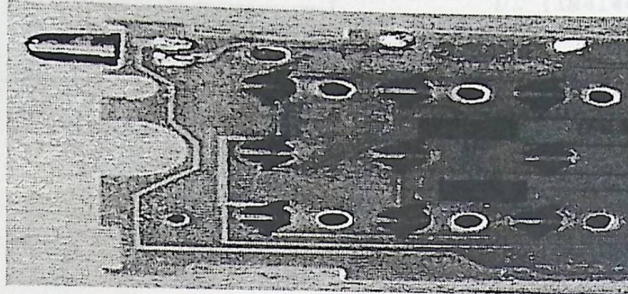


Figure 4.4: The Circuit Board

At the end of the circuit board there is an infrared LED, or Light Emitting Diode as figure 4.4. You can think of an LED as a small light bulb. Many LEDs produce visible light, but a remote's LED produces infrared light that is invisible to the human eye. It is not invisible to all eyes, however. For example, if you have a camcorder it can see the infrared light. Point your remote at the camera and push a button. You will be able to see the infrared light flashing in the viewfinder. The receptor in the IR Receiver is able to see infrared light as well.

4.2.2 IR Receiver unit:

The main aim of the receiver unit as shown in figure 4.5 is to receive the signal that sent from the transmitter. In the receiving stages there are many operation that will occur .Photo detector in the receiver sense the IR signal and generate electrical signal, amplified , filtered and sampled then entered to computer (to microprocessor) that process this signal then perform actions that related to this signal

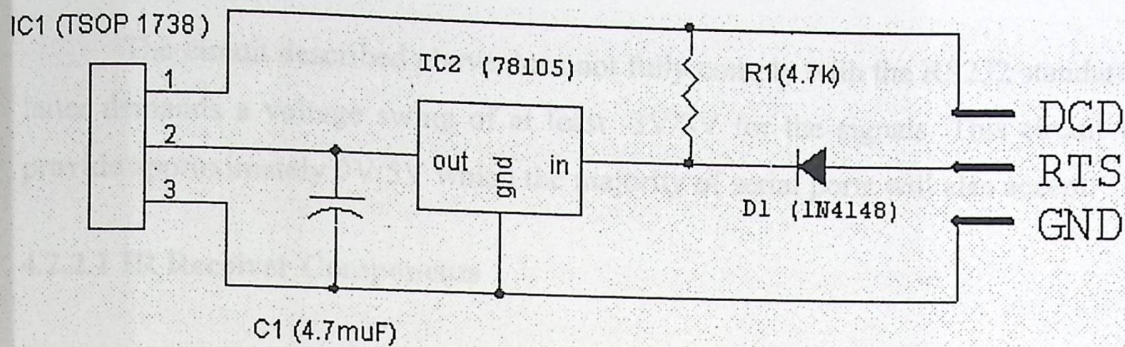


Figure 4.5: Infrared Receiver Circuit

The power of the receiver circuit comes from the RTS line of the serial port. The diode protects the circuit from the negative voltage that is usually on the RTS pin. The software driver will change the serial port settings on initialization so that the circuit will get the necessary positive voltage. For most standard PC serial ports this will be approximately 10V. The voltage regulator will convert the input voltage to exactly 5V. As the input voltage for the 78Lxx type regulators has to be at least 2V higher than the output voltage to work reliably. Instead of pulling the power supply from the serial port you can also use a different source like e.g. the USB port or the internal 5V line to power the circuit. Then you won't need a voltage regulator at all.

The capacitor is optional but recommends using it anyway to enhanced filter process. The resistor is a pull-up that makes sure the DCD line is on a valid voltage level if the receiver does not receive anything. When an infrared signal is detected TSOP 1738 (this is true for active low receivers, if you use an active high receiver leave away resistor or make it a pull-down to GND) pulls DCD down to ground, which should already be interpreted as a logical "1" (DCD=0) by the serial port.

The circuit described above does not fully comply with the RS232 standard as latter demands a voltage swing of at least -3V/3V for the signals. This circuit will provide approximately 0V/5V which the majority of serial ports will also accept.

4.2.2.1 IR Receiver Components

The IR Receiver circuit consists of several components, as shown below:

4.2.2.1.1 TSOP 1738 IR receiver:

The TSOP1738 Series are miniaturized receivers for infrared remote control systems. PIN diode and preamplifier are assembled on lead frame, the epoxy package is designed as IR filter. The demodulated output signal can directly be decoded by a microprocessor. TSOP17... Is the standard IR remote control receiver series, supporting all major transmission codes Figure (4.6) shows this chip.

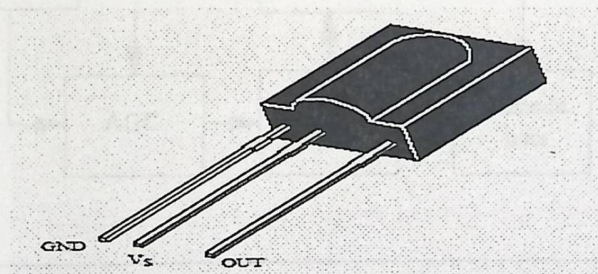


Figure 4 .6: TSOP Pins

□ Features

- Photo detector and preamplifier in one package
- Internal filter for PCM frequency
- Improved shielding against electrical field disturbance
- TTL and CMOS compatibility
- Output active low
- Low power consumption
- High immunity against ambient light
- Continuous data transmission possible (up to 2400 bps)
- Suitable burst length .10 cycles/burst

□ Block Diagram

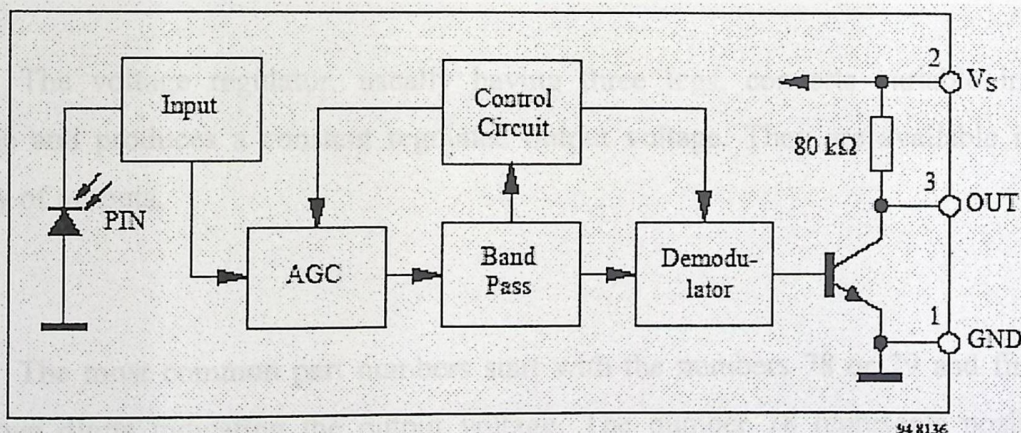


Figure 4.7: TSOP Block Diagram

□ TSOP limitations:

The data signal should fulfill the following conditions

- The carrier signal frequency should be close to the center frequency about of band pass (38 KHZ).
- burst length should be 10 cycles/burst or longer
- Each burst must be between 10 cycles and 70 cycles a gap time of at least 14 cycles is necessary, when a disturbance signal is applied to the TSOP it can still receive the data signal. However the sensitivity is reduced to that level that no unexpected pulses will occur.

4.2.2.1.2 Voltage Regulator 78L05

The voltage regulator, usually having three legs, converts variable input voltage and produces a constant regulated output voltage. They are available in a variety of outputs.

The most common part numbers start with the numbers 78 or 79 and finish with two digits indicating the output voltage. The number 78 represents positive voltage and 79 negative one. The 78XX series of voltage regulators are designed for positive input. And the 79XX series is designed for negative input.

The receiver needs a positive regulator in order to convert the input voltage from the 12V serial port to +5v.

The voltage regulator protects itself not only with current limiting, but also with a thermal sensor that prevents damage from excessive power dissipation an overload that could occur even though the current alone remained below the limiting value.

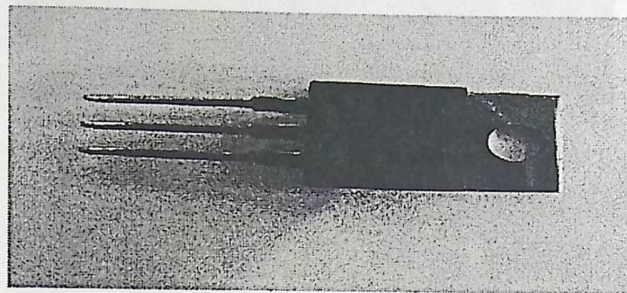


Figure 4.8 Single voltage regulator

4.2.2.1.3 1N4148 universal diode

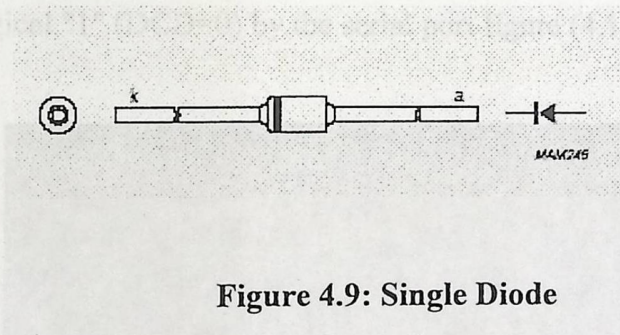
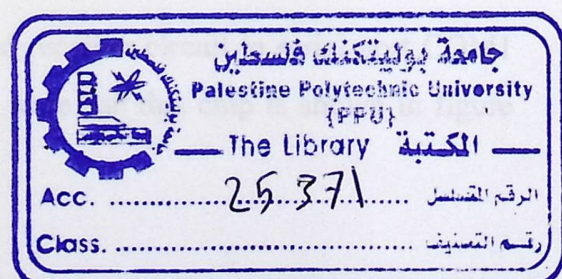


Figure 4.9: Single Diode

This chip is used to conduct current in one direction and block current in the other direction and convert AC to DC figure (4.9) shows this chip



4.2.2.1.4 Electrolytic capacitor

The capacitor we used is about 4.7 μ F as shown in figure (4.10).

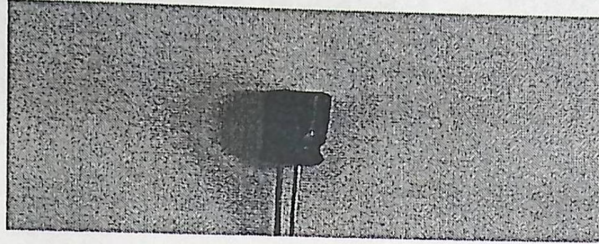


Figure 4.10: Single capacitor

4.2.2.1.5 Resistor

Is a pull-up that makes sure the DCD line is on a valid voltage level if the receiver does not receive anything. When an infrared signal is detected by TSOP (this is true for active low receivers, if we use an active high receiver leave away resistor or make it a pull-down to GND) pulls DCD down to ground, which should already be interpreted as a logical "1" (DCD=0) by the serial port figure (4.11) shows this chip.

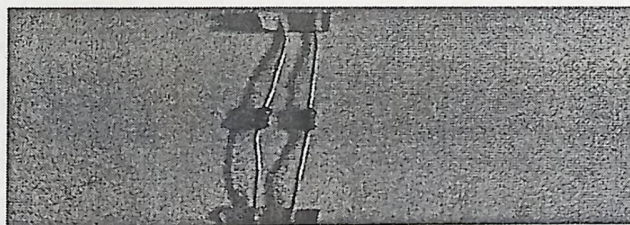


Figure 4.11: Resistors

4.2.2.1.6 Female connector

This chip is used to connect the infrared receiver circuit to serial port COM1 or COM2 it is with nine pins only pin (1,5,7) were use this chip is shown in figure 4.12.

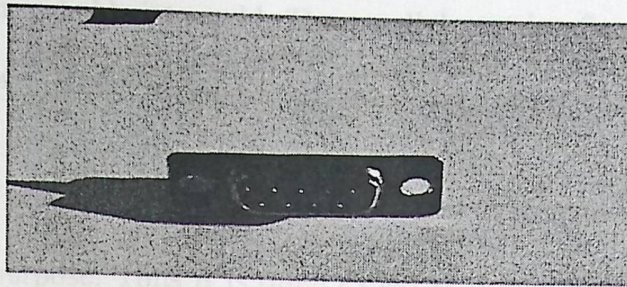


Figure 4.12: Single Female connectors

4.2.3 Control Unit

The system need control unit to achieve all the operation of the design. The control unit represented by two main parts; the first one represented by computer that connect the units of project together by serial and parallel ports such as the IR receiver unit connected with computer through serial port and the application unit connected with it by parallel port.

The second part represented by software driver programmed using visual basic .net that install on computer to fetch the signals from serial port then decode and analyze it after that execute and perform the specific task refer to it (we will discuss this part in detailed at chapter five).

Now we will illustrate the ports that connect the project subsystem.

4.2.3.1 Parallel ports

Parallel port is a simple and inexpensive tool for building computer controlled devices and projects. The simplicity and ease of programming makes parallel port

popular in electronics hobbyist world. The parallel port is often used in Computer controlled robots, Atmel/PIC programmers, home automation, etc.

The primary use of parallel port is to connect printers to computer and is specifically designed for this purpose. Thus it is often called as printer Port or Centronics port (this name came from a popular printer manufacturing company 'Centronics' who devised some standards for parallel port). You can see the parallel port connector in the rear panel of your PC. It is a 25 pin female (DB25) connector (to which printer is connected). On almost all the PCs only one parallel port is present, but you can add more by buying and inserting ISA/PCI parallel port cards.

□ **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

Hardware

The pin outs of DB25 connector is shown in the figure 4.13.

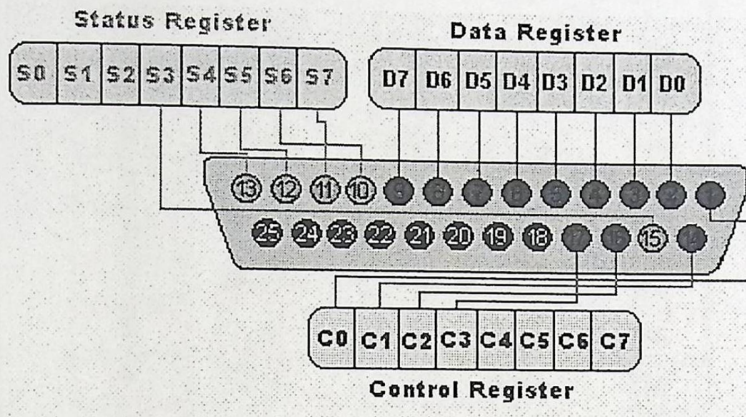


Figure 4.13: Parallel Port Register

The lines in DB25 connector are divided in to three groups, they are

- Data lines (data bus)
- Control lines
- 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 computer through Status lines. These lines are connected to Data, Control and Status registers internally. The details of parallel port signal lines are given in table (4.1)

Table 4.1: parallel port signal line

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	-	-	-

□ Parallel port registers

As you know, 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 standard parallel port are:

- Data register
- Status registers
- Control register

As there names specifies, 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 what ever writes to these registers, will appear in corresponding lines as voltages, by measure it with a millimeter. And What ever 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.

□ Where these registers are?

In an IBM PC, these registers are IO mapped and will have unique address. We have to find these addresses to work with 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 registers in this manner. The table (4.2) below shows the register addresses of LPT1 and LPT2.

Table 4.2: register addresses of LPT1 and LPT2

Register	LPT1	LPT2
data register(base address + 0)	0x378	0x278
status register (base address + 1)	0x379	0x279
control register (base address + 2)	0x37a	0x27a

4.2.3.2 Serial ports

We have to use The Serial Port on the computer receives asynchronous data at suitable speeds for our data. This, however, requires an extra hardware/logic interface between the IR receiver and the port. Some of the main incompatible features between the data links were. The serial port expects asynchronous data, while the remote control sends self-timed data. The serial port can only accept words of length 5 or 7 bits. It also expects start and stop bits. The remote control has 12 bit words with a header. To overcome these incompatibilities, we need hardware such as a demodulator and logic that encodes the IR signal as 5 bit / 7 bit words perhaps sending 1 data signal from the remote control at a time. Thus, a word received on the serial port would represent the data signal equivalent to a time period of T in the remote control pulse code. The serial port has to be used to connect the infrared receiver circuit through it by serial port cable as shown in figure 4.14.

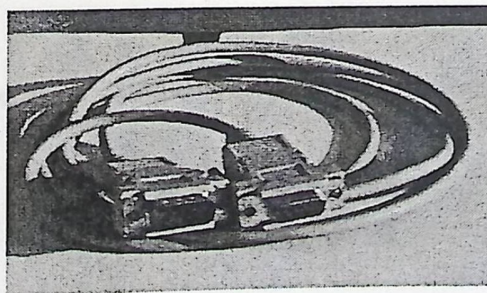


Figure 4.14: Serial port cable

The serial port is an I/O (Input/Output) device. An I/O device is just a way to get data into and out of a computer. There are many types of I/O devices such as serial ports, parallel ports, disk drive controllers, Ethernet boards, universal serial buses, etc. Most PC's have one or two serial ports. Each has a 9-pin connector (sometimes 25-pin) on the back of the computer. Computer programs can send data (bytes) to the transmit pin (output) and receive bytes from the receive pin (input). The other pins are for control purposes and ground.

The serial port is much more than just a connector. It converts the data from parallel to serial and changes the electrical representation of the data. Inside the computer, data bits flow in parallel (using many wires at the same time). Serial flow is a stream of bits over a single wire (such as on the transmit or receive pin of the serial connector). For the serial port to create such a flow, it must convert data from parallel (inside the computer) to serial on the transmit pin and conversely.

□ Pins and Wires:

Old PC's used 25 pin connectors but only about 9 pins were actually used so today most connectors are only 9-pin. Each of the 9 pins usually connects to a wire. Besides the two wires used for transmitting and receiving data, another pin (wire) is signal ground. The voltage on any wire is measured with respect to this ground. Thus the minimum number of wires to use for 2-way transmission of data is 3. Except that it has been known to work with no signal ground wire but with degraded performance and sometimes with errors. As shown in figure (4.15).



Figure (4.15): Serial port pins

There are still more wires which are for control purposes (signaling) only and not for sending bytes. All of these signals could have been shared on a single wire, but instead, there is a separate dedicated wire for every type of signal. Some (or all) of these control wires are called "modem control lines". Modem control wires are either in the asserted state (on) of +12 volts or in the negated state (off) of -12 volts. One of these wires is to signal the computer to stop sending bytes out the serial port cable. Conversely, another wire signals the device attached to the serial port to stop sending bytes to the computer. If the attached device is a modem, other wires may tell the modem to hang up the telephone line or tell the computer that a connection has been made or that the telephone line is ringing (someone is attempting to call in).

□ Data flow

Data (bytes representing letters, pictures, etc.) flows into and out of your serial port. Flow rates (such as 56k (56000) bits/sec) are (incorrectly) called "speed". But almost everyone says "speed" instead of "flow rate".

It's important to understand that the average speed is often less than the specified speed. Waits (or idle time) result in a lower average speed. These waits may include long waits of perhaps a second due to flow control. At the other extreme there may be very short waits (idle time) of several micro-seconds between bytes. If the device on the serial port (such as a modem) can't accept the full serial port speed, then the average speed must be reduced.

After this, specific software needs to be written in order to decode / record the pulse code and interpret it as application commands. This software can be also part of the application.

PC Com Port - EIA-574 RS-232/V.24 pin out on a DB-9 pin used for asynchronous data shown in figure (4.16).

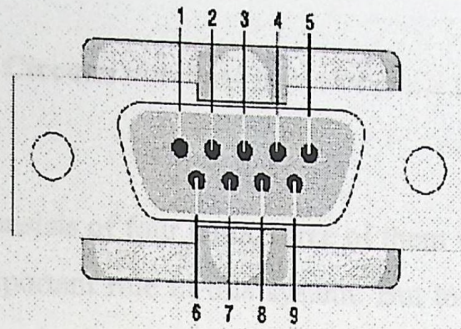


Figure (4.16): PC Com Port

Table 4.3:Serial Port Pins

Pin	Signal
1	Data Carrier detector
2	Received Data
3	Transmitted Data
4	Data Terminal Ready
5	Signal Ground
6	Data Set Ready
7	Request To Send
8	Clear To Send
9	Ring Inductor

4.2.3 Application unit

The application unit of the system consists mainly from home automation system and the external circuits that interface this system.

relay is an electro-magnetic switch which is useful if you want to use a circuit to switch on and off a light bulb (or anything else) connected to a power supply.

Figure 4.18 shows a typical relay (with "normally-open" contacts).

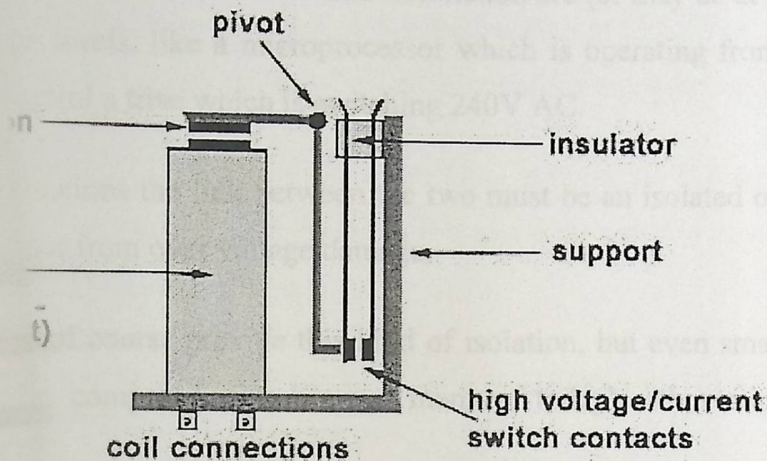


Figure 4.18: Relay Parts

ded to operate the relay coil is more than can be supplied by the power source (e.g., a battery), so a transistor is usually needed.

The main point of a relay is to use a small amount of power in the control circuit to switch a much larger amount of power. For example, a small relay can be energized using 5 volts and 50 milliamps, but the armature can support 120V AC at 2 amps (240 watts).

4.2.4.1 Interfacing Circuits

The system consists of four parts, three of them were explained previously, and each part has an important role in this system. But to achieve the system objectives and operates as one unit, there's a need to integrate each unit with others through the interfacing circuits .The interfacing circuits are:

4.2.4.1.1 Relay

A relay, quite simply, is a small machine consisting of an electromagnet (coil), a switch, and a spring. The spring holds the switch in one position, until a current is passed through the coil. The coil generates a magnetic field which moves the switch. It's that simple. We can use a very small amount of current to activate a relay, and the switch can often handle a lot of current, shows figure 4.17.

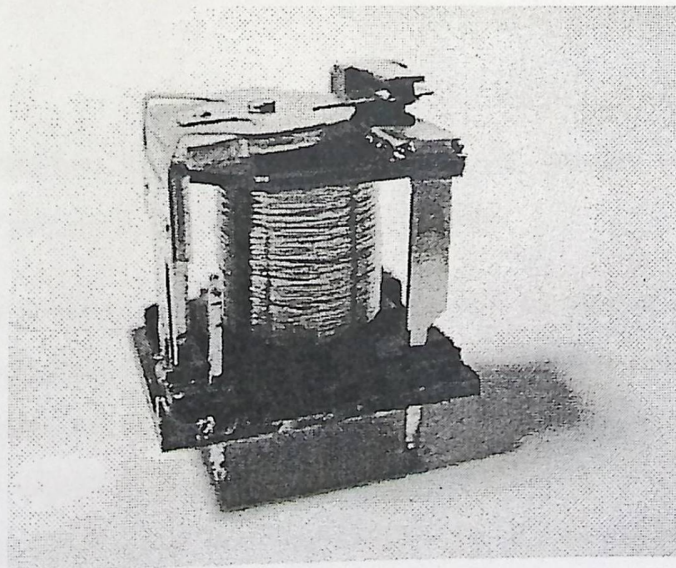


Figure 4.17: Relay

And the relay is an electro-magnetic switch which is useful if you want to use a low voltage circuit to switch on and off a light bulb (or anything else) connected to the 220v mains supply.

The figure (4.18) shows a typical relay (with "normally-open" contacts).

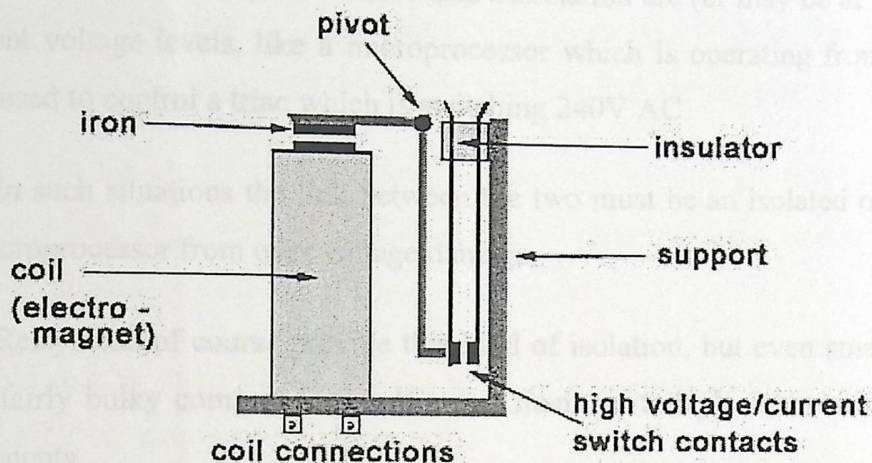


Figure 4.18: Relay Parts

The current needed to operate the relay coil is more than can be supplied by most chips (op. amps etc), so a transistor is usually needed.

In general, the point of a relay is to use a small amount of power in the electromagnet coming from a small dashboard switch or a low-power electronic circuit to move an armature that is able to switch a much larger amount of power. For example, if we want the electromagnet to energize using 5 volts and 50 milliamps (250 mill watts), while the armature can support 120V AC at 2 amps (240 watts).

4.2.4.1.2 Optocaplars

There are many situations where signals and data need to be transferred from one subsystem to another within a piece of electronics equipment, or from one piece of equipment to another, without making a direct electrical connection.

Often this is because the source and destination are (or may be at times) at very different voltage levels, like a microprocessor which is operating from 5V DC but being used to control a triac which is switching 240V AC.

In such situations the link between the two must be an isolated one, to protect the microprocessor from over voltage damage.

Relays can of course provide this kind of isolation, but even small relays tend to be fairly bulky compared with ICs and many of today's other miniature circuit components.

Because they're electro-mechanical, relays are also not as reliable, and only capable of relatively low speed operation. Where small size, higher speed and greater reliability are important, a much better alternative is to use an optocoupler.

These use a beam of light to transmit the signals or data across an electrical barrier, and achieve excellent isolation.

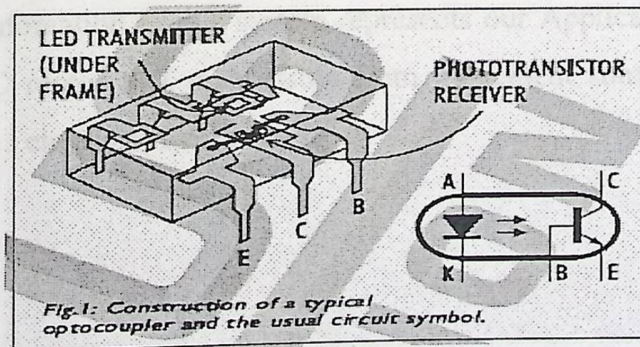


Figure 4.19: Optocouplers

Optocouplers typically come in a small 6-pin or 8-pin IC package, but are essentially a combination of two distinct devices: an optical transmitter, typically a gallium arsenide LED (light-emitting diode) and an optical receiver such as a phototransistor or light-triggered diac.

The two are separated by a transparent barrier which blocks any electrical current flow between the two, but does allow the passage of light. The basic idea is shown in Fig above, along with the usual circuit symbol for an optocoupler.

Usually the electrical connections to the LED section are brought out to the pins on one side of the package and those for the phototransistor or diac to the other side, to physically separate them as much as possible.

This usually allows optocouplers to withstand voltages of anywhere between 500V and 7500V between input and output.

Optocouplers are essentially digital or switching devices, so they're best for transferring either on/off control signals or digital data. Analog signals can be transferred by means of frequency or pulse-width modulation.

4.2.4.2 Home Automation

The home automation in this system represents our Application unit, and this home contains DC motor that responsible from movement some objects such as (Doors, Windows ...etc), Fan, Lampetc.

4.2.4.2.1 DC motor circuit

DC motor circuit responsible of movement objects, and this circuit needs two relay to run the motor in two direction (left and right) and works properly, the input of this circuit connected to parallel port with (D0) for relay1 and (D1) for relay2, when the user press the (button 0) in remote control that's mean the motor will move to the right, and the motor will move to the left if you press (button 1) and you can prevent the motor from rotate by press in (button 4).

The figure 4.20 illustrate how does the DC motor circuit work

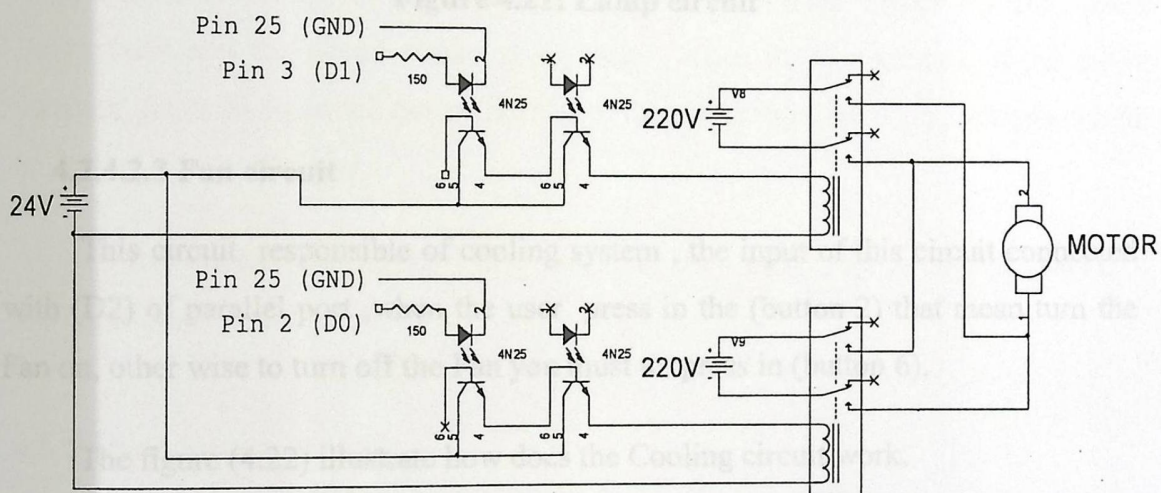


Figure 4.20: DC motor circuit

4.2.4.2.2 Lamp circuit

This circuit responsible of lighting system, the input of this circuit connected with (D3) of parallel port, when the user press in the (button 3) that mean turn the lamp on, other wise to turn off the lamp you must to press in (button 7).

The figure 4.21 illustrate how does the lighting circuit work.

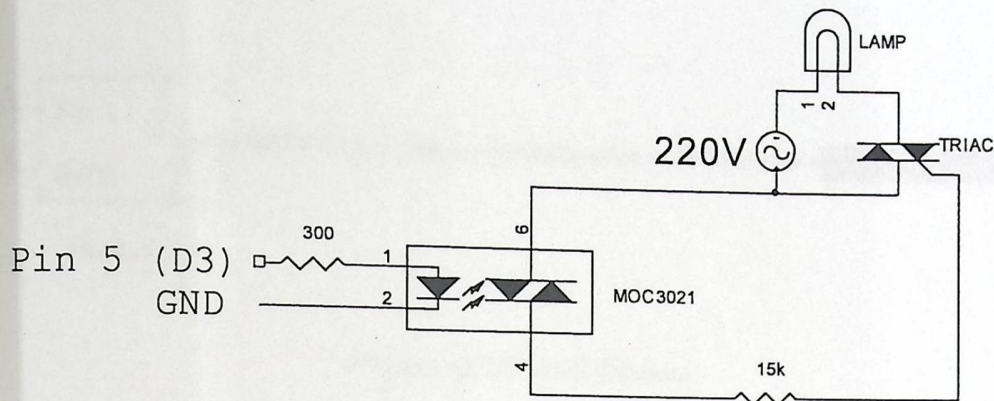


Figure 4.21: Lamp circuit

4.2.4.2.3 Fan circuit

This circuit responsible of cooling system , the input of this circuit connected with (D2) of parallel port ,when the user press in the (button 2) that mean turn the Fan on, other wise to turn off the Fan you must to press in (button 6).

The figure (4.22) illustrate how does the Cooling circuit work.

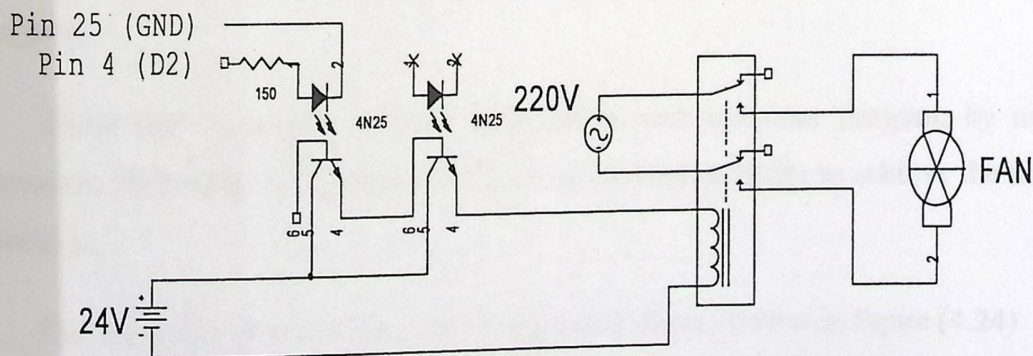


Figure 4.22: Fan circuit

4.2.4.2.4 Heat Sensor Circuit

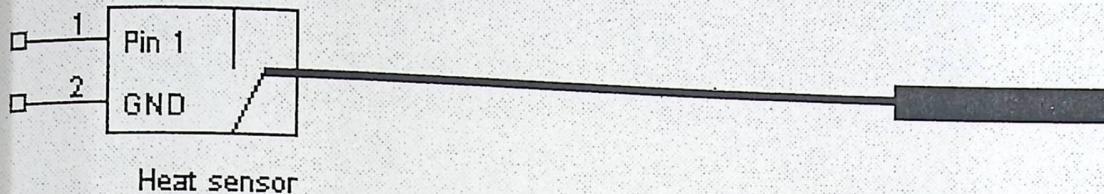


Figure 4.23 Heat Sensor

The heat sensor circuit considered as a feed back circuit to the motor temperature and the circuit works as a switch , when the temperature of the motor increase above or equal the determined level the sensor turn on ,it connect ground pin to pin 1 of the parallel port.

The change of the signal 0 or 1 that read from pin 1 represents the change of the temperature level

4.3 Overall System Design

The system design represents the complete interface between the units designed.

There for, to control a home automation and computer program by remote control the following design had been built and tested carefully to achieve the needed objectives.

The complete characteristic of the system is shown below in figure (4.24)

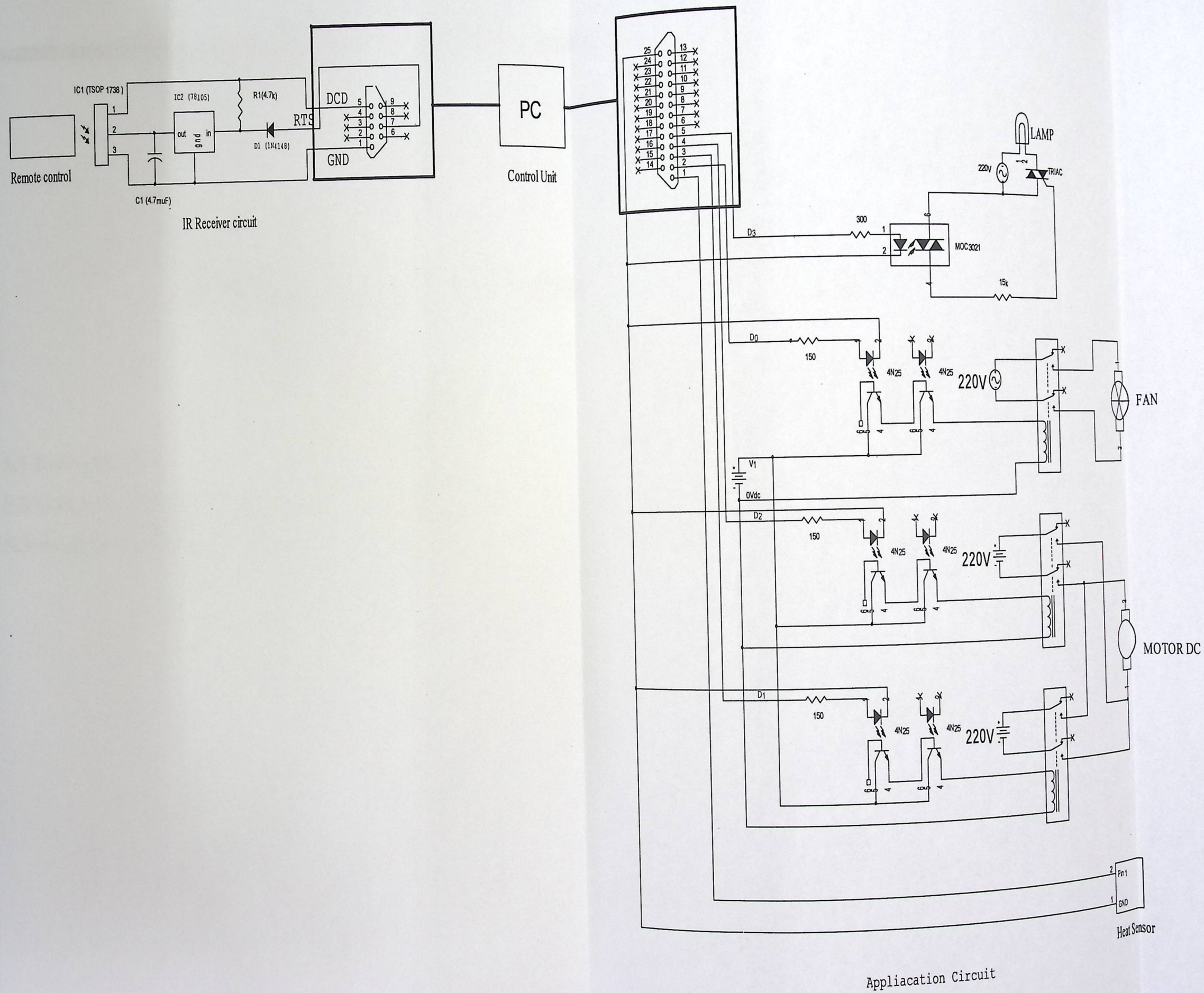


Figure 4.24: The System Design Circuit

5.1 Introduction

In this chapter we are going to describe the software system design and to describe some of the user interface and hardware details.

Software System Design

And it contains the flowchart for all processes in this project from first step in the project (initialize the ports & receiving the signal) to the final step (perform the action related to the signal).

The software is programming in VB.net.

5.1 Introduction

5.2 General System Flowchart

5.3 System Operational Flowchart

- The general flowchart for the system can be described in following steps:
- The remote control will send signals to the receiver via an infrared light.
- The receiver will receive those signals, and send them to the PC.
- The PC takes the signal and processes them to perform the action related with it. For more details see the flowchart (figure 5.1)

Chapter Five

Software System Design

5.1 Introduction

In this chapter we are going to describe the software system design and to describe some of the using methods and algorithm design as followed.

And it contains the flowchart for all processes in this project from first step in the project (initialize the ports & receiving the signal) to the final step (perform the action related to the signal).

The overall software is programming in VB.net.

5.2 General System Flowchart

- The general flowchart for the system can be described in following steps:
- The remote control will send signals to the receiver via an infrared light.
- The receiver will receive these signals, and send them to the PC.
- The PC takes the signal and processes them to perform the action related with it. For more details see the flowchart (figure 5.1).

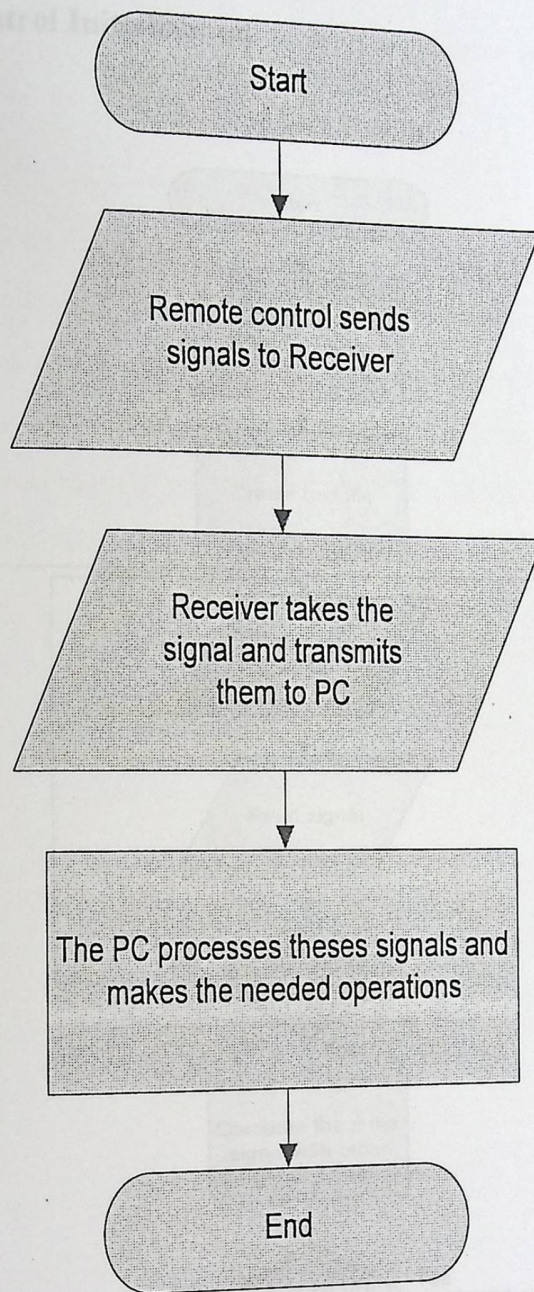


Figure 5.1 General System Flowchart

5.3 System Operational Flowchart

In this section we will show the system operational flowchart as following.

5.3.1 Remote Control Initialization

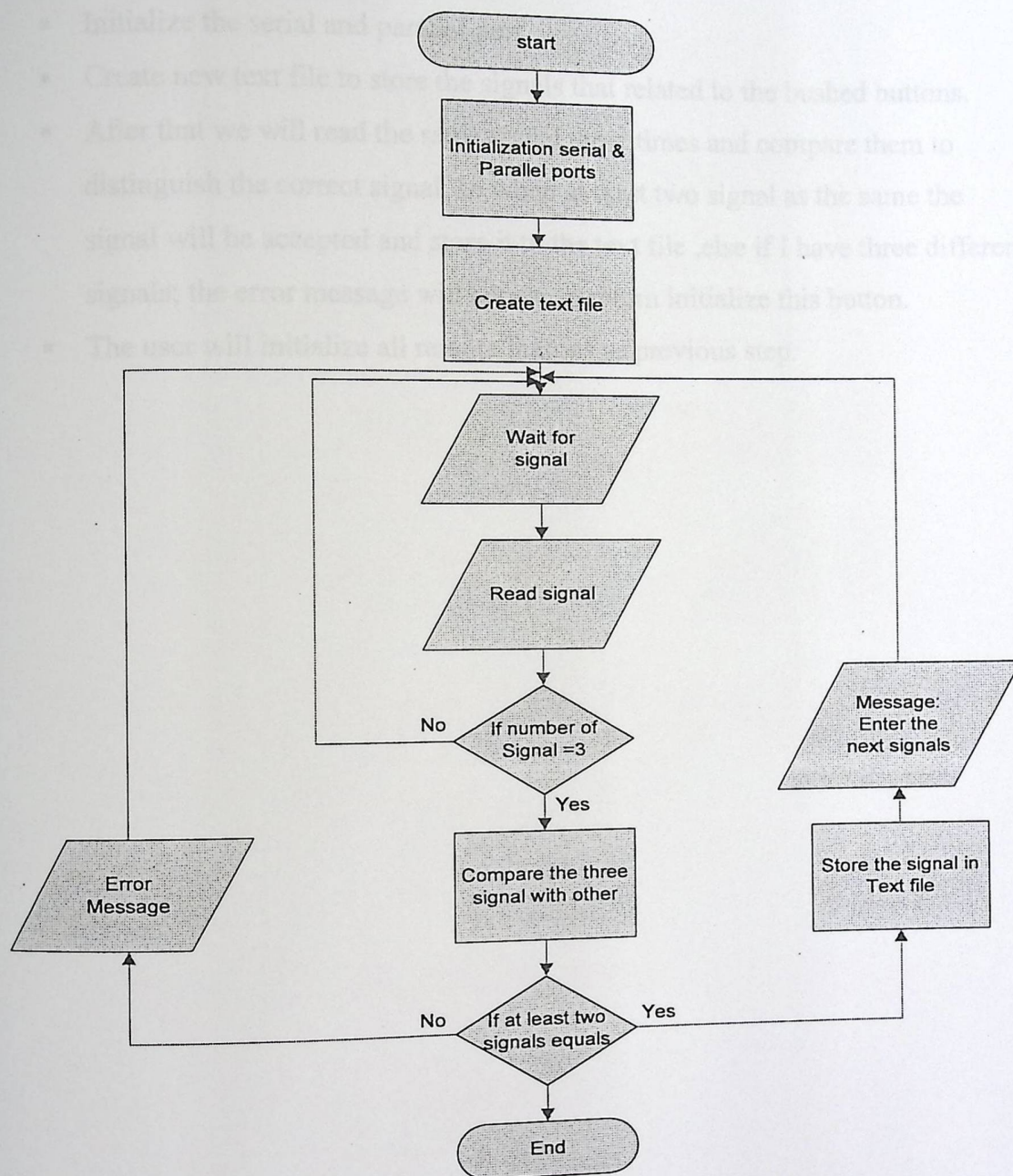


Figure 5.2 Remote Control Initialization flowchart

As shown in previous (figure 5.2) the remote control initializations consist of many steps such as:

- Initialize the serial and parallel port.
- Create new text file to store the signals that related to the bushed buttons.
- After that we will read the same signal three times and compare them to distinguish the correct signal, so when at least two signal as the same the signal will be accepted and store it to the text file ,else if I have three different signals; the error message will appear to return initialize this button.
- The user will initialize all remote buttons as previous step.

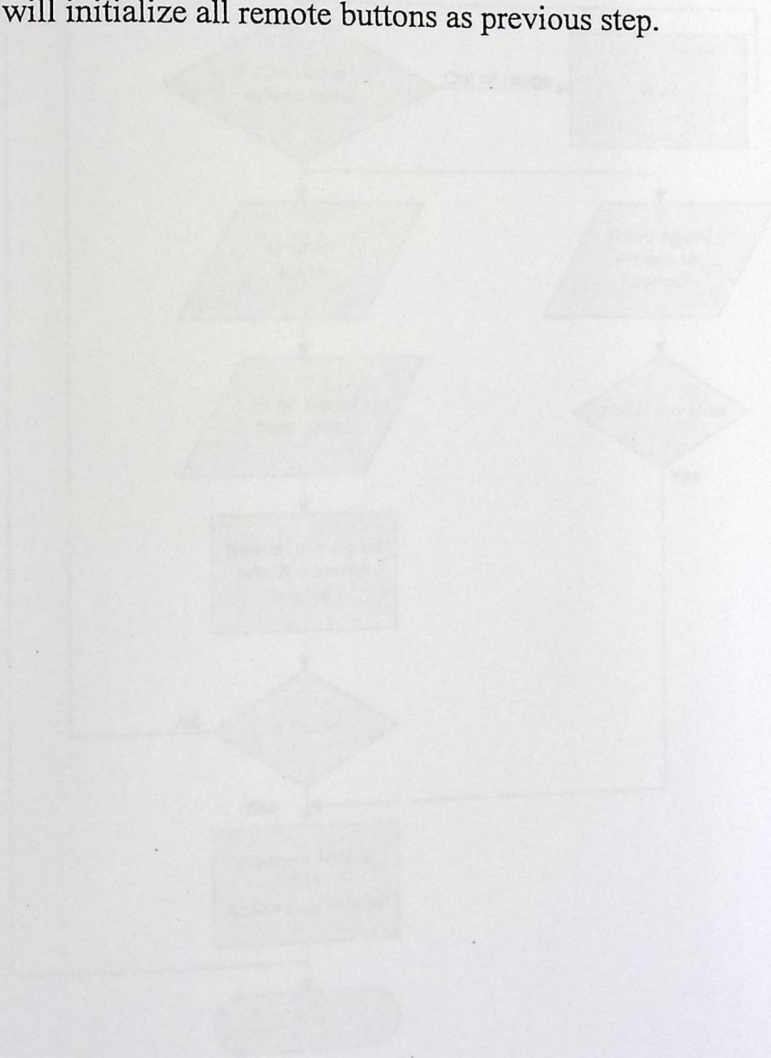


Figure 5.3: Remote Signal Operation Flowchart

5.3.2 Remote Signals Operations

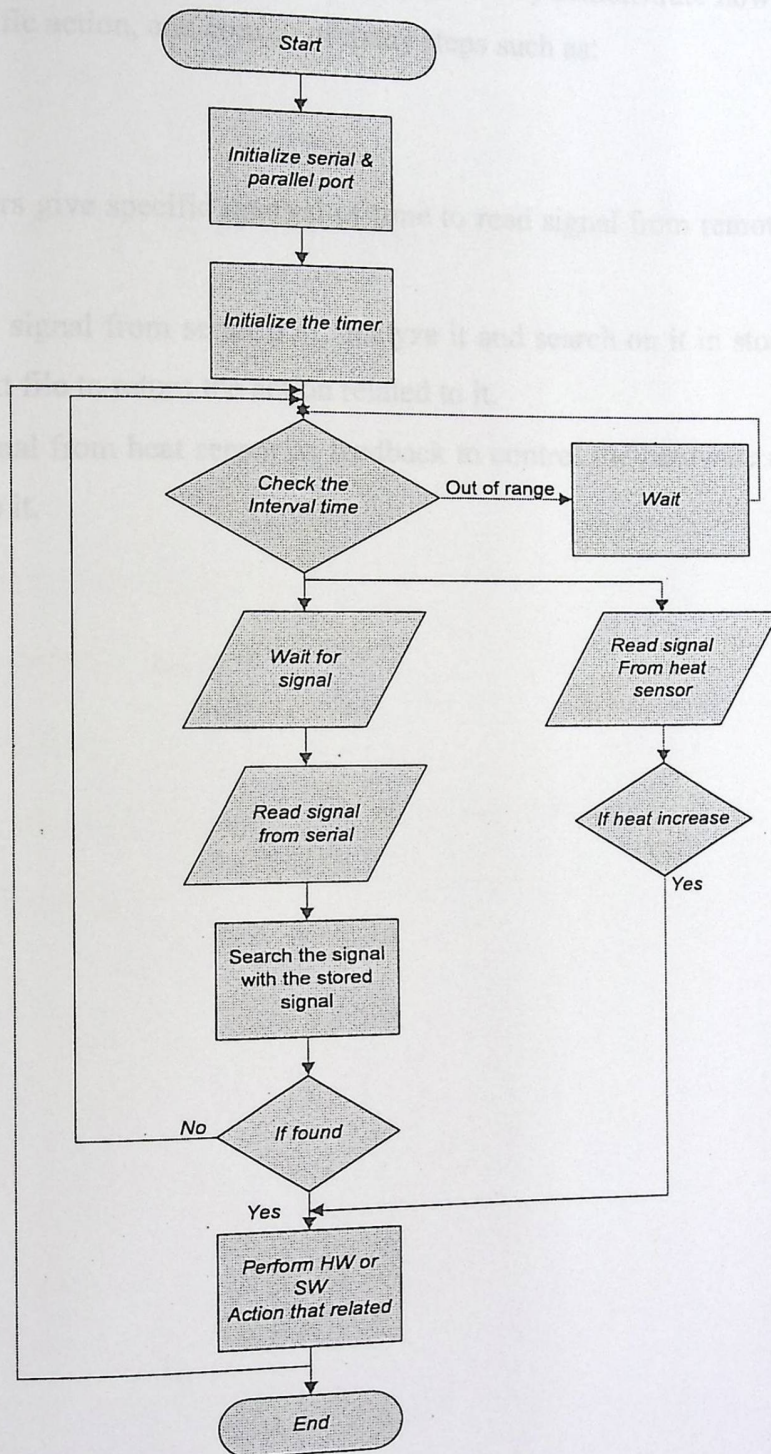


Figure 5.3: Remote Signals Operations Flowchart

As shown in previous (figure 5.3) the remote control has many signal which perform specific operation, and the previous (figure 5.3) demonstrate how the signal perform its specific action, and consist of many steps such as:

- The timers give specific interval of time to read signal from remote and heat sensor.
- Read the signal from serial port, analyze it and search on it in stored signals in the text file to return the action related to it.
- Read signal from heat sensor as feedback to control the hardware application related to it.

Implementation and Testing

6.1 Introduction

Implementation and Testing

In this chapter,

The implementation and testing was done using the following tools and components:

6.1 Introduction

6.2 Implementation

6.3 Testing

6.4 Implementation and testing for Hardware component.

6.2 Implementation

The implementation process is done synchronized with the testing operation, because each implementation phase will take many testing to ensure that there are no errors, to trace the implementation phases see 6.4.

Chapter Six

Implementation and Testing

6.1 Introduction

In this chapter we are going to show the implementation and testing processes for our system.

The implementation and testing was done using the following tools and components:

- Connectors with different colors.
- IC stands.
- 10*20 cm broad board.
- All the ICs that are depicted in the design chapter (see chapter 4).
- Wrapper tool for wrapping the connectors on the ICs stands.
- A wire grabber and a wire cutter.
- A digital millimeter for testing.
- Oscilloscope.

6.2 Implementation

The implementation process is done synchronized with the testing operation, because each implementation phase will take many testing to ensure that there are no errors, to trace the implementation phases see 6.4.

6.3 Testing

In this section we will demonstrate the testing of each subsystem separately.

6.3.1 Testing the receiver circuit

Receiver circuit used to convert infrared light to electrical signal that reads by serial port as in figure 6.1.

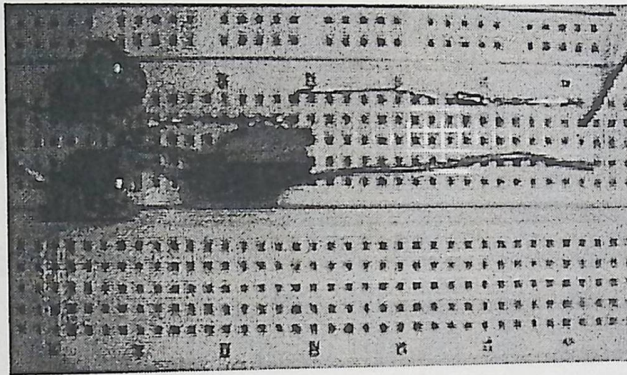


Figure 6.1: Infrared Receiver circuit

We test this circuit by C program language that reads the signal and store it, this circuit had been tested many times to emphasize better results, and we use this program to test this circuit.


```

#include<dos.h>
#include<stdio.h>
#include<conio.h>
#include<string.h>
#define port 0x3F8
#define tt 30000
int as[20],ac[20],ch;
void main(void)
{
clrscr();
outportb(port+4,0x0B);
for(int d=0;d<1;d++)
{
for(int e=0;e<20;e++)
ac[e]=5;
/*****
int c=1;
int a[tt],q=0,i,h=0,t=-1,r[50],f[50],t1[50];
for(i=1;i<50;i++){ r[i]=5;t1[5];}
while(c!= 0x00)
{
c=inport(port+6);
c=c & 0x80;
if(c==0x00)
{
for(i=1;i<tt;i++)
{ // delay(1);
c=inport(port+6);
c=c & 0x80;
a[i]=c;
} //end for
c=0x00;
} //end if
} //while loop
a[0]=0x00;
////////// array //////////
int w;

for(w=1;w<tt;w++)
{
if(a[w-1]==a[w]) q++;
else{
r[t++]=q;
t1[t]=a[w-1];
q=1;
}
}
}

```



```

    h++;
}
}

/////////////////////////////////
int a1,a2,ra,u=0;
a1=150;a2=450;
if(a1>a2) ra=150;
else ra=150;
for(i=0;i<50;i++)
{
    if((((a2)-ra)<=r[i]) && (r[i]<=((a2)+ra)))
    { f[i]=1;}
    else {f[i]=0;}
}

for(i=0;i<50;i++)
{
    printf("%d(%d) ",r[i],f[i]);
}

/////////////////////////////////
int j=0,o=0;
for(i=0;i<20;i++)
as[i]=0;
for(i=1;i<50;i++)
{
    if(f[i]!=1) j++;
    else{as[o]=j;j=0;o++;}
}
printf("\n");
for(i=0;i<20;i++)
{
    printf("%d ",as[i]);
}

/////////////////////////////////
h=0;
for(i=0;i<50;i++)
h=h+r[i];

delay(250);
/*****/

/*****/
FILE *fp;

```



```

fp = fopen("c:\\omar.txt", "w+");
for(int ii=2;ii<10;ii++)
fprintf(fp,"%d",as[ii] );
fclose(fp);
/*****

```

```

}

```

```

getch();

```

```

}

```

When we press any key we will see this result

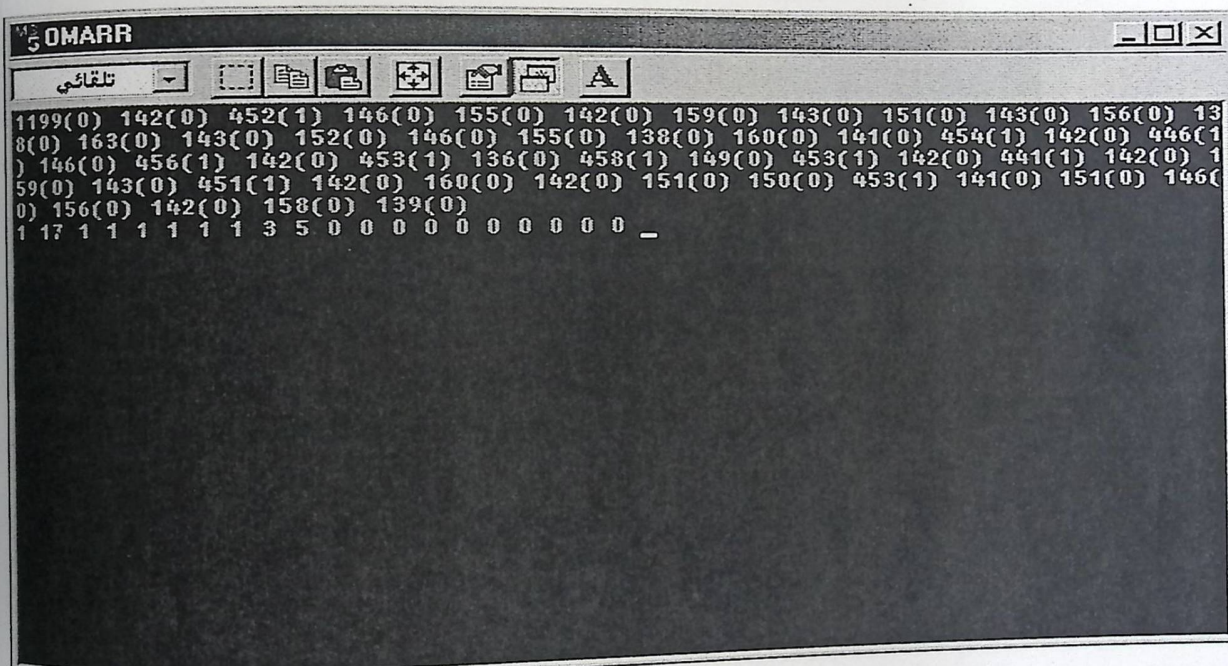


Figure 6.2: CMD screen

The value in the parentheses represent the signal, so that level 0 is mean zero volt and 1 is mean 12 volt ,and the number before each parentheses is represent the length of the pulse, we used the pulse width to decode the signals.

6.3.2 Testing the opt couplers and relays

This circuit was built to control the direction of the motor and turn on the fan, It consists of 3-relays 24V of each with 4N25 opt coupler with necessary resistors.

The control signal comes from parallel port to opt coupler on the other side of the opt coupler transistor take 24V from DC transformer to switch the relay as figure 6.3.

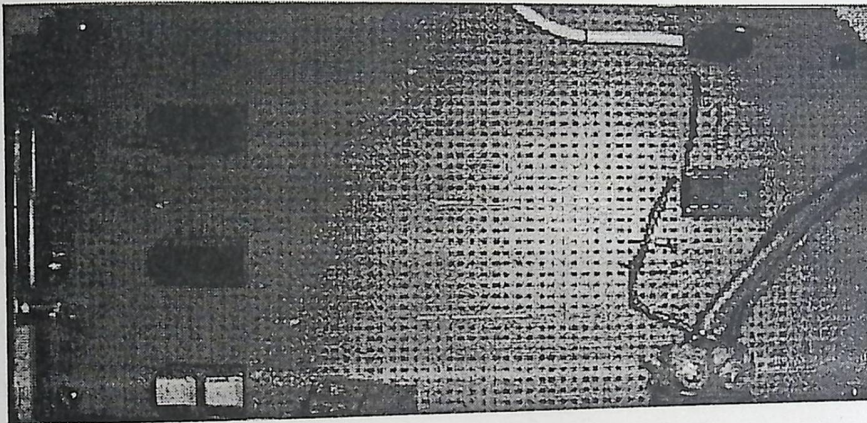


Figure 6.3: Relay Circuit

This circuit tested by C program and operated usefully, the program is:-

```
int main(void)
{
    outportb(0x378, 0x03);
    return 0;
}
```


6.3.3 Testing the temperature circuit

The temperature circuit was built to sense the temperature of the motor environment as in figure 6.4.

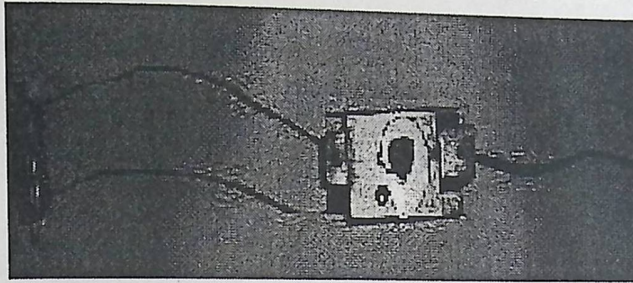


Figure 6.4: Temperature Sensor

When temperature increase at some level the sensor send signal to pin1 of parallel port, the program reads this pin continuously when there a signal in pin1 the program indicate that the temperature increase.

This circuit had been tested many times with C language to emphasize good results. The C program is:

```
int main(void)
{
    int result;
    result = inportb(0x37A);
    result=result & 0x01 ;
    if(result==1)
        printf("\n The heat increas \n");
    else
        printf("\n The heat not increas \n");
    return 0;
}
```


6.3.4 Testing the program with the receiver circuit and application

The completed program will be in the appendices that read the signal from serial port and perform software action and hardware action by parallel port.

This program is tested with receiver circuit and application and work well, but we notice from our testing that the C program works much better and faster than VB.net with parallel port.

6.3.5 Access the serial and parallel port using VB.net

To access the serial and parallel port using the VB.net we get an input.dll library, which must add to system32, and then we define a parallel port and serial port that will be used as an object to open and access the ports, as the followed:

```
Option Strict Off
Option Explicit On
Imports System.IO
Imports System.Threading.ThreadPriority
```

```
Friend Class Form1
    Inherits System.Windows.Forms.Form
```

```
    Public Declare Auto Function CreateFile Lib "kernel32.dll" _
        (ByVal lpFileName As String, ByVal dwDesiredAccess As Int32, _
        ByVal dwShareMode As Int32, ByVal lpSecurityAttributes As IntPtr, _
        ByVal dwCreationDisposition As Int32, ByVal dwFlagsAndAttributes As
        Int32, _
        ByVal hTemplateFile As IntPtr) As IntPtr
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles MyBase.Load
```

```
        Out(Val("&H378"), Val("&H00"))
        Dim hSerialPort, hParallelPort As IntPtr
```



```
hSerialPort = CreateFile("COM1", &H80000000 Or &H40000000, 0,  
IntPtr.Zero, _  
3, &H80, IntPtr.Zero)
```

```
End Sub  
End class
```

6.4 Implementation and testing for Hardware component

As we mentioned previously after completing any circuit that we described it in chapter four we use the C language to test each subsystem and test every input and output with different cases and speed.

The testing operation is done in more than one phases as following:

Phase 1:

The receiver circuit is build and tested using external voltage source and millimeters and oscilloscope after this phase we ensuring all receiver components are operated will.

Phase 2:

The receiver circuit connected to the serial port and we programming with C language to read signals from receiver circuit, and the result is the program read signal from receiver circuit.

Phase 3:

The optcoupler circuit is build and tested with external circuit is build and tested with external voltage source; it's tested and operated successfully.

Phase 4:

The optocoupler circuit is connected to the parallel port and provide with external DC voltage transformer to provide voltage, and connected to relays and tested the circuit is operate successfully.

Phase 5:

The receiver circuit and opt coupler circuit with relays connected with serial and parallel port consecutively and perform the related action on relays with parallel port.

Phase 6:

After completing VB.net program with test the hole system on the program, the finally results are successfully on software and hardware control.

7

Conclusion and Future Work

7.1 Introduction

7.2 Conclusion

7.3 Future view

Chapter Seven

Conclusion and Future Work

7.1 introduction

The personal computer based general purpose control system using general remote control using infrared receiver circuit has achieved the main ideas prepared to .it is now ready to be applied on the pc. The main feature that achieved is:-

- 1- the team enable to control the hardware applications connected with pc from far distance approximately 15 meters
- 2- The team enable to construct the personal computer based general purpose control system using general remote control with reasonable cost.
- 3- The team enables to make the system ease to use by everyone.
- 4- Controlling the hardware applications connected with pc through a parallel port, the team enables to send infrared signal from remote control to IR receiver circuit connected with pc through serial port, then analyze the signal and send the related voltage signal to the parallel port to run the application related to that signal.
- 5- Controlling the windows API

7.2 Conclusions

After the team finish the design and implementation of the project and integrated overall system.

The team concludes that:

- There is ability to use infrared light in controlling applications connected with PC remotely by using infrared circuit that convert IR light to voltage signal can be read it through serial port.
- The user can use remote technique to run several applications on computer directly with out need to use keyboard or mouse, instead of that we uses remote button to run several applications.
- The user can use remote to control of any hardware application connected with computer ports such as open and close dooretc.
- Remote technology available in computer field so any one can use and run this technology with out difficulties and provide software need to run remote circuit under computer area.
- design general purpose remote control with lowest costs.
- General purpose remote control make life easier.

7.3 Future view

During the system period the team enables to achieve the main goals he started to do them. although there are sum techniques that could be used to reach the objects in less time but we started our system from the zero and we used the most available and simple equipments to emphasize the result and prove this new technology of controlling the hardware application connected with pc by infrared light.

There are some media that can be used as an advance for the system, these media will be our recommendation to help other students achieve more results .So in our system we have used the serial port and parallel ports to connect our subsystem with pc, so we assume the future work make an USB interfacing for our subsystem, on the other hand making the control of PC through the internet.

Appendices

Appendix A: Software Driver Code.

Appendix B: Datasheets.

Appendix A: Software Driver Code.


```

1  '/******This Code represents The source Code Of Software Driver*****//
2  '/******Software Driver*****//
3  '/******Option Strict Off*****//
4  Option Strict Off
5  Option Explicit On
6  'Imports package
7  Imports System.IO
8  Imports System.Threading.ThreadPriority
9
10 Friend Class Form1
11     Inherits System.Windows.Forms.Form
12     '/// Function to read and write from registers
13     Public Declare Auto Function CreateFile Lib "kernel32.dll"
14         (ByVal lpFileName As String, ByVal dwDesiredAccess As _Int32,
15          ByVal dwShareMode As Int32, ByVal lpSecurityAttributes As IntPtr,
16          ByVal dwCreationDisposition As Int32, ByVal dwFlagsAndAttributes
17          As Int32, _
18          ByVal hTemplateFile As IntPtr) As IntPtr
19 #Region "Windows Form Designer generated code "
20     Public Sub New()
21         MyBase.New()
22         If m_vb6FormDefInstance Is Nothing Then
23             If m_InitializingDefInstance Then
24                 m_vb6FormDefInstance = Me
25             Else
26                 Try
27                     'For the start-up form, the first instance created is the
28                     default instance.
29                     If System.Reflection.Assembly.GetExecutingAssembly.EntryPoint.
30                     DeclaringType Is Me.GetType Then
31                         m_vb6FormDefInstance = Me
32                     End If
33                 Catch
34                 End Try
35             End If
36         End If
37         'This call is required by the Windows Form Designer.
38         InitializeComponent()
39     End Sub
40     'Form overrides dispose to clean up the component list.
41     Protected Overrides Sub Dispose(ByVal Disposing As Boolean)
42         If Disposing Then
43             If Not components Is Nothing Then
44                 components.Dispose()
45             End If
46         End If
47         MyBase.Dispose(Disposing)
48     End Sub
49     'Required by the Windows Form Designer
50     Private components As System.ComponentModel.IContainer
51     'NOTE: The following procedure is required by the Windows Form Designer
52     'It can be modified using the Windows Form Designer.
53     'Do not modify it using the code editor.
54     Friend WithEvents Label1 As System.Windows.Forms.Label
55     Friend WithEvents Label2 As System.Windows.Forms.Label
56     Friend WithEvents Label3 As System.Windows.Forms.Label
57     Friend WithEvents Button2 As System.Windows.Forms.Button
58     Friend WithEvents Button3 As System.Windows.Forms.Button
59     Friend WithEvents ProgressBar1 As System.Windows.Forms.ProgressBar
60     Friend WithEvents ProgressBar2 As System.Windows.Forms.ProgressBar
61     Friend WithEvents ProgressBar3 As System.Windows.Forms.ProgressBar
62     Friend WithEvents Button1 As System.Windows.Forms.Button
63     Friend WithEvents TextBox1 As System.Windows.Forms.TextBox
64     Friend WithEvents NotifyIcon1 As System.Windows.Forms.NotifyIcon
65     Friend WithEvents ContextMenu1 As System.Windows.Forms.ContextMenu
66     Friend WithEvents MenuItem1 As System.Windows.Forms.MenuItem
67     Friend WithEvents MenuItem2 As System.Windows.Forms.MenuItem
68     Friend WithEvents Timer1 As System.Windows.Forms.Timer
69     Private WithEvents Timer1 As System.Windows.Forms.Timer
70     Private Sub InitializeComponent()
71     <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
72         Me.components = New System.ComponentModel.Container
73         Dim resources As System.Resources.ResourceManager = New System.Resources.
74         ResourceManager(GetType(Form1))
75         Me.Label1 = New System.Windows.Forms.Label

```



```

1  '/* *****This Code represents The source
2  '/* *****
3  '/* *****
4  Option Strict Off
5  Option Explicit On
6  'Imports package
7  Imports System.IO
8  Imports System.Threading.ThreadPriority
9
10 Friend Class Form1
11     Inherits System.Windows.Forms.Form
12     '/// Function to read and write from registry
13     Public Declare Auto Function CreateFile Lib "kernel32.dll" (ByVal lpFileName As String, ByVal dwShareMode As Int32, ByVal dwCreationDisposition As Int32, ByVal hTemplateFile As IntPtr) As IntPtr
14     As Int32, _
15     ByVal hTemplateFile As IntPtr) As IntPtr
16
17 #Region "Windows Form Designer generated code"
18     Public Sub New()
19         MyBase.New()
20         If m_vb6FormDefInstance Is Nothing Then
21             If m_InitializingDefInstance Then
22                 m_vb6FormDefInstance = Me
23             Else
24                 Try
25                     'For the start-up form, the first instance created is the
26                     default instance.
27                     If System.Reflection.Assembly.GetExecutingAssembly.EntryPoint.DeclaringType Is Me.GetType Then
28                         m_vb6FormDefInstance = Me
29                     End If
30                 Catch
31                     End Try
32             End If
33         End If
34         'This call is required by the Windows Form Designer.
35         InitializeComponent()
36     End Sub
37     'Form overrides dispose to clean up the component.
38     Protected Overrides Sub Dispose(ByVal disposing As Boolean)
39         If disposing Then
40             If Not components Is Nothing Then
41                 components.Dispose()
42             End If
43         End If
44         MyBase.Dispose(disposing)
45     End Sub
46     'Required by the Windows Form Designer
47     Private components As System.ComponentModel.IContainer
48     'NOTE: The following procedure is required by the Windows Form Designer.
49     'It can be modified using the Windows Form Designer.
50     'Do not modify it using the code editor.
51     Friend WithEvents Label1 As System.Windows.Forms.Label
52     Friend WithEvents Label2 As System.Windows.Forms.Label
53     Friend WithEvents Label3 As System.Windows.Forms.Label
54     Friend WithEvents Button2 As System.Windows.Forms.Button
55     Friend WithEvents Button3 As System.Windows.Forms.Button
56     Friend WithEvents ProgressBar1 As System.Windows.Forms.ProgressBar
57     Friend WithEvents ProgressBar2 As System.Windows.Forms.ProgressBar
58     Friend WithEvents ProgressBar3 As System.Windows.Forms.ProgressBar
59     Friend WithEvents Button1 As System.Windows.Forms.Button
60     Friend WithEvents TextBox1 As System.Windows.Forms.TextBox
61     Friend WithEvents NotifyIcon1 As System.Windows.Forms.NotifyIcon
62     Friend WithEvents ContextMenu1 As System.Windows.Forms.ContextMenu
63     Friend WithEvents MenuItem1 As System.Windows.Forms.MenuItem
64     Friend WithEvents MenuItem2 As System.Windows.Forms.MenuItem
65     Private WithEvents Timer1 As System.Windows.Forms.Timer
66     <System.Diagnostics.DebuggerStepThrough>
67     Me.components = New System.ComponentModel.Container()
68     Dim resources As System.Resources.ResourceManager = New System.Resources.ResourceManager(GetType(Form1))
69     Me.Label1 = New System.Windows.Forms.Label

```



```

70 Me.Label2 = New System.Windows.Forms.Label
71 Me.Label3 = New System.Windows.Forms.Label
72 Me.Button2 = New System.Windows.Forms.Button
73 Me.Button3 = New System.Windows.Forms.Button
74 Me.ProgressBar1 = New System.Windows.Forms.ProgressBar
75 Me.ProgressBar2 = New System.Windows.Forms.ProgressBar
76 Me.ProgressBar3 = New System.Windows.Forms.ProgressBar
77 Me.Button1 = New System.Windows.Forms.Button
78 Me.TextBox1 = New System.Windows.Forms.TextBox
79 Me.NotifyIcon1 = New System.Windows.Forms.NotifyIcon(Me.components)
80 Me.ContextMenu1 = New System.Windows.Forms.ContextMenu
81 Me.MenuItem1 = New System.Windows.Forms.MenuItem
82 Me.MenuItem2 = New System.Windows.Forms.MenuItem
83 Me.Timer1 = New System.Windows.Forms.Timer(Me.components)
84 Me.SuspendLayout()
85 '
86 'Label1
87 '
88 Me.Label1.BackColor = System.Drawing.Color.White
89 Me.Label1.Enabled = False
90 Me.Label1.Location = New System.Drawing.Point(32, 176)
91 Me.Label1.Name = "Label1"
92 Me.Label1.Size = New System.Drawing.Size(88, 24)
93 Me.Label1.TabIndex = 2
94 Me.Label1.Text = "First Signal"
95 Me.Label1.TextAlign = System.Drawing.ContentAlignment.MiddleCenter
96 '
97 'Label2
98 '
99 Me.Label2.BackColor = System.Drawing.SystemColors.ActiveCaptionText
100 Me.Label2.Enabled = False
101 Me.Label2.Location = New System.Drawing.Point(32, 216)
102 Me.Label2.Name = "Label2"
103 Me.Label2.Size = New System.Drawing.Size(88, 24)
104 Me.Label2.TabIndex = 4
105 Me.Label2.Text = "Secound Signal"
106 Me.Label2.TextAlign = System.Drawing.ContentAlignment.MiddleCenter
107 '
108 'Label3
109 '
110 Me.Label3.BackColor = System.Drawing.SystemColors.ActiveCaptionText
111 Me.Label3.Enabled = False
112 Me.Label3.Location = New System.Drawing.Point(32, 256)
113 Me.Label3.Name = "Label3"
114 Me.Label3.Size = New System.Drawing.Size(88, 24)
115 Me.Label3.TabIndex = 5
116 Me.Label3.Text = "Thired Signal"
117 Me.Label3.TextAlign = System.Drawing.ContentAlignment.MiddleCenter
118 '
119 'Button2
120 '
121 Me.Button2.Enabled = False
122 Me.Button2.Location = New System.Drawing.Point(40, 296)
123 Me.Button2.Name = "Button2"
124 Me.Button2.Size = New System.Drawing.Size(80, 40)
125 Me.Button2.TabIndex = 6
126 Me.Button2.Text = "Clear"
127 '
128 'Button3
129 '
130 Me.Button3.Enabled = False
131 Me.Button3.Location = New System.Drawing.Point(32, 120)
132 Me.Button3.Name = "Button3"
133 Me.Button3.Size = New System.Drawing.Size(88, 32)
134 Me.Button3.TabIndex = 7
135 Me.Button3.Text = "Add"
136 '
137 'ProgressBar1
138 '
139 Me.ProgressBar1.Enabled = False
140 Me.ProgressBar1.Location = New System.Drawing.Point(136, 176)
141 Me.ProgressBar1.Name = "ProgressBar1"
142 Me.ProgressBar1.Size = New System.Drawing.Size(240, 24)

```



```

143 Me.ProgressBar1.TabIndex = 8
144 '
145 'ProgressBar2
146 '
147 Me.ProgressBar2.Enabled = False
148 Me.ProgressBar2.Location = New System.Drawing.Point(136, 216)
149 Me.ProgressBar2.Name = "ProgressBar2"
150 Me.ProgressBar2.Size = New System.Drawing.Size(240, 24)
151 Me.ProgressBar2.TabIndex = 9
152 '
153 'ProgressBar3
154 '
155 Me.ProgressBar3.Enabled = False
156 Me.ProgressBar3.Location = New System.Drawing.Point(136, 256)
157 Me.ProgressBar3.Name = "ProgressBar3"
158 Me.ProgressBar3.Size = New System.Drawing.Size(240, 24)
159 Me.ProgressBar3.TabIndex = 10
160 '
161 'Button1
162 '
163 Me.Button1.Font = New System.Drawing.Font("Arial", 8.25!, System.Drawing.
FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
164 Me.Button1.Location = New System.Drawing.Point(32, 16)
165 Me.Button1.Name = "Button1"
166 Me.Button1.Size = New System.Drawing.Size(144, 32)
167 Me.Button1.TabIndex = 3
168 Me.Button1.Text = "intialization"
169 '
170 'TextBox1
171 '
172 Me.TextBox1.BackColor = System.Drawing.Color.FromArgb(CType(128, Byte),
CType(128, Byte), CType(255, Byte))
173 Me.TextBox1.Enabled = False
174 Me.TextBox1.Location = New System.Drawing.Point(136, 128)
175 Me.TextBox1.Name = "TextBox1"
176 Me.TextBox1.Size = New System.Drawing.Size(176, 20)
177 Me.TextBox1.TabIndex = 12
178 Me.TextBox1.Text = ""
179 '
180 'NotifyIcon1
181 '
182 Me.NotifyIcon1.ContextMenu = Me.ContextMenu1
183 Me.NotifyIcon1.Icon = CType(resources.GetObject("NotifyIcon1.Icon"),
System.Drawing.Icon)
184 Me.NotifyIcon1.Text = "Remot"
185 Me.NotifyIcon1.Visible = True
186 '
187 'ContextMenu1
188 '
189 Me.ContextMenu1.MenuItems.AddRange(New System.Windows.Forms.MenuItem() {Me
.MenuItem1, Me.MenuItem2})
190 Me.ContextMenu1.RightToLeft = System.Windows.Forms.RightToLeft.No
191 '
192 'MenuItem1
193 '
194 Me.MenuItem1.Index = 0
195 Me.MenuItem1.Text = "show window"
196 '
197 'MenuItem2
198 '
199 Me.MenuItem2.Index = 1
200 Me.MenuItem2.Text = "Exit"
201 '
202 'Timer1
203 '
204 Me.Timer1.Enabled = True
205 Me.Timer1.Interval = 20
206 '
207 'Form1
208 '
209 Me.AutoScaleBaseSize = New System.Drawing.Size(6, 13)
210 Me.BackColor = System.Drawing.SystemColors.ActiveBorder
211 Me.BackgroundImage = CType(resources.GetObject("$this.BackgroundImage"),

```



```

System.Drawing.Image)
Me.ClientSize = New System.Drawing.Size(616, 414)
Me.Controls.Add(Me.TextBox1)
Me.Controls.Add(Me.ProgressBar3)
Me.Controls.Add(Me.ProgressBar2)
Me.Controls.Add(Me.ProgressBar1)
Me.Controls.Add(Me.Button3)
Me.Controls.Add(Me.Button2)
Me.Controls.Add(Me.Label3)
Me.Controls.Add(Me.Label2)
Me.Controls.Add(Me.Button1)
Me.Controls.Add(Me.Label1)
Me.Cursor = System.Windows.Forms.Cursors.Default
Me.Font = New System.Drawing.Font("Arial", 8.25!, System.Drawing.FontStyle
.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.Location = New System.Drawing.Point(4, 23)
Me.Name = "Form1"
Me.RightToLeft = System.Windows.Forms.RightToLeft.No
Me.Text = "Form1"
Me.ResumeLayout(False)

```

End Sub

#End Region

#Region "Upgrade Support "

```

Private Shared m_vb6FormDefInstance As Form1
Private Shared m_InitializingDefInstance As Boolean
Public Shared Property DefInstance() As Form1

```

Get

```

If m_vb6FormDefInstance Is Nothing OrElse m_vb6FormDefInstance.
IsDisposed Then

```

```

    m_InitializingDefInstance = True
    m_vb6FormDefInstance = New Form1
    m_InitializingDefInstance = False

```

End If

DefInstance = m_vb6FormDefInstance

End Get

Set(ByVal Value As Form1)

m_vb6FormDefInstance = Value

End Set

End Property

#End Region

#Region "API fonction"

```

'//function to execute keyboard keys functions
Declare Function keybd_event Lib "user32" Alias "keybd_event" _
(ByVal bVk As Byte, ByVal bScan
As Byte, ByVal dwFlags As Long, ByVal dwExtraInfo As Long) As Long

```

#End Region

```

'// datatype declaration
Public acc As Integer = -1
Public Ali As String() = New String(28) {"TAB", "ENTER", "SHIFT", "ALT",
"ESCAPE", "WIN", "SPACE", "LEFT ARROW", "UP ARROW", "RIGTH ARROW", "DOWN ARROW",
", "DELETE", "F5", "WIN+D", "WIN+L", "ALT+F4", "ALT+F", "Ctrl+Alt+del", "D0",
"D1", "D2", "D3", "D4", "D5", "D6", "D7", "end program", "RMOUSE", "HELP"}
Public Sal As Byte() = New Byte(28) {&H9, &HD, &H10, &H12, &H1B, &H5B, &H20, &
H25, &H26, &H27, &H28, &H2E, &H74, &H44, &H4C, &H73, &H46, &H0, &H1, &H2, &H4,
&H8, &H10, &H20, &H40, &H80, &H0, &H45, &H0}
Public valu As Int64() = New Int64(28) {}
Public output As Byte = &H0
Public index As Integer = 0
Public t2 As Integer = 5000
Public t3 As Integer = 75
Public a1 As Integer
Public a2 As Integer
Public hidel As Integer = 0
Public helptest As Boolean = False
Public help1 As New Form2

```

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MyBase.Load

```



```

176 '/// initialization serial and parralell ports
177 Out(Val("&H378"), Val("&H00"))
178 Dim hSerialPort, hParallelPort As IntPtr
179 hSerialPort = CreateFile("COM1", &H80000000 Or &H40000000, 0, IntPtr.Zero,
- 3, &H80, IntPtr.Zero)

```

```

80 End Sub
81
82

```

```

83 Private Sub Label1_Click(ByVal sender As System.Object, ByVal e As System.
84 EventArgs) Handles Label1.Click

```

```

85 End Sub
86

```

```

87 Private Sub Button1_Click_1(ByVal sender As System.Object, ByVal e As System.
88 EventArgs) Handles Button1.Click

```

```

'*****
Label1.Enabled() = True
Label2.Enabled() = True
Label3.Enabled() = True
TextBox1.Enabled() = True
ProgressBar1.Enabled = True
ProgressBar2.Enabled = True
ProgressBar3.Enabled = True
Button2.Enabled() = True
Button3.Enabled() = True
'*****

```

```

89 End Sub
90
91

```

```

92 Private Sub Label3_Click(ByVal sender As System.Object, ByVal e As System.
93 EventArgs) Handles Label3.Click

```

```

94 End Sub
95

```

```

96 Private Sub ProgressBar1_Click(ByVal sender As System.Object, ByVal e As
97 System.EventArgs) Handles ProgressBar1.Click

```

```

98 End Sub
99

```

```

100 Private Sub Label4_Click(ByVal sender As System.Object, ByVal e As System.
101 EventArgs)

```

```

102 End Sub
103

```

```

104 '/// function to inititalize the remote signal of each key of the remote and
105 store it in the text file
106 Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.
107 EventArgs) Handles Button3.Click

```

```

acc = acc + 1
TextBox1.Text = Ali(acc)
ProgressBar1.Value() = 0
ProgressBar2.Value() = 0
ProgressBar3.Value() = 0

```

```

Dim aq As Boolean
Dim aw As Int64 = 0

```

```

'*****

```

```

Dim ae As Integer
Dim ax1 As Int64
Dim ax2 As Int64
Dim ax3 As Int64

```

```

For ae = 0 To 2
'*****
Dim i As Int64
Dim str1 As String
Dim x As Integer = 0
Dim index As Integer = 0
Dim counter As Integer = 0

```



```
Dim y1, y2 As Integer
```

```
Dim a As Integer() = New Integer(t2) {}  
Dim count As Integer() = New Integer(t3) {}  
Dim tcount As Integer() = New Integer(t3) {}  
Dim tcount2 As Integer() = New Integer(t3) {}  
Dim tcount3 As Integer() = New Integer(t3) {}  
Dim j, t1 As Integer
```

```
For j = 0 To t3  
    tcount(j) = 5  
Next
```

```
x = Str(Inp(Val("&H3FE")))
```

```
While x = 128  
    x = Str(Inp(Val("&H3FE")))  
End While
```

```
' Read 10000 sample
```

```
For i = 0 To t2  
    Out(Val("&H3FC"), Val("&H0B"))  
    x = Str(Inp(Val("&H3FE")))  
    x = (x And 128)  
    'stor samles in array  
    a(i) = x  
Next
```

```
Dim DELA As Integer = 0
```

```
y1 = a(0)
```

```
For i = 1 To t2  
    y2 = a(i)  
    If y2 = y1 Then  
        counter += 1  
    Else  
        count(index) = counter  
        If y1 = 0 Then  
            tcount(index) = y1  
        Else  
            tcount(index) = 1  
        End If  
        index += 1  
        counter = 0  
    End If  
    y1 = y2  
Next
```

```
'*****
```

```
t1 = 0
```

```
For j = 2 To t3  
    If (tcount(j) = 1 And ((count(j) > (90 - 40)) And (count(j) < (90 + 40)))) Then
```

```
        t1 = t1 + 1
```

```
        tcount2(t1) = 1
```

```
    ElseIf (tcount(j) = 1 And ((count(j) > (30 - 40)) And (count(j) < (30 + 40)))) Then
```

```
        t1 = t1 + 1
```

```
        tcount2(t1) = 0
```

```
    End If
```

```
Next
```

```
'*****
```

```
y1 = tcount2(0)
```

```
index = 0
```

```
counter = 0
```

```
For i = 1 To t1
```

```
    y2 = tcount2(i)
```

```
    If y2 = y1 Then
```

```
        counter += 1
```

```
    Else
```

```
        tcount3(index) = counter
```

```
        index += 1
```

```
        counter = 0
```

```
    End If
```



```

        y1 = y2

    Next

    str1 = " "
    '*****

    For i = 0 To index
        str1 = str1 & System.Convert.ToString(tcount3(i))
    Next
    If ae = 0 Then
        ax1 = Val(str1)
        ProgressBar1.Value() = 100
    ElseIf ae = 1 Then
        ax2 = Val(str1)
        ProgressBar2.Value() = 100
    ElseIf ae = 2 Then
        ax3 = Val(str1)
        ProgressBar3.Value() = 100
    End If
    delay()
Next
Dim file1 As New System.IO.StreamWriter("c:\test.txt", True)
If ax1 = ax2 Then
    file1.WriteLine(ax1)
    MessageBox.Show(" Inter The next signal", "signal")
ElseIf ax1 = ax3 Then
    file1.WriteLine(ax1)
    MessageBox.Show(" Inter The next signal", "signal")
ElseIf ax2 = ax3 Then
    file1.WriteLine(ax2)
    MessageBox.Show(" Inter The next signal", "signal")
Else : acc = acc - 1
    MessageBox.Show(" Signal is wrong ", "signal")
End If

    file1.Close()
End Sub
'//create text file
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles Button2.Click
    Dim file1 As New System.IO.StreamWriter("c:\test.txt")
    file1.Close()
    ProgressBar1.Value() = 0
    ProgressBar2.Value() = 0
    ProgressBar3.Value() = 0
    acc = -1
End Sub
'// initialize the timer with 20ms
Private Sub InitializeTimer()

    Timer1.Interval = 20

    Timer1.Enabled = True

End Sub
Public Sub MenuItem1_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MenuItem1.Click
    Timer1.Enabled = False
    Me.Show()
End Sub
Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MenuItem2.Click
    End
End Sub
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles Timer1.Tick
    Hide()
    Dim Start, Finish, TotalTime As Double
    '/////////////////////////*****

```



```

31 start = Microsoft.VisualBasic.DateAndTime.Timer
32 finish = start + 0.001 ' Set end time for 0.001 -second duration.
33 Do While Microsoft.VisualBasic.DateAndTime.Timer < finish
34     '// execute all this statement while (0.001 -second duration)not
35 finished
36     If (Inp(Val("&H37A"))) And &H1 = &H1 Then '// if heat increase
37         output = (output And &HF9)
38         Out(Val("&H378"), output)
39     End If
40
41     Dim aq As Integer = -1
42     Dim aw As Int64 = 0
43     '// read from text file
44     Dim file As New System.IO.StreamReader("c:\test.txt")
45     For aq = 0 To 28
46         valu(aq) = Val(file.ReadLine)
47     Next
48
49     file.Close()
50     Dim i As Int64
51     Dim str1 As String
52     Dim x As Integer = 0
53
54     Dim counter As Integer = 0
55     Dim y1, y2 As Integer
56
57     Dim a As Integer() = New Integer(t2) {}
58     Dim count As Integer() = New Integer(t3) {}
59     Dim count1 As Integer() = New Integer(t3) {}
60     Dim tcount As Integer() = New Integer(t3) {}
61     Dim tcount2 As Integer() = New Integer(t3) {}
62     Dim tcount3 As Integer() = New Integer(t3) {}
63     Dim j, t1 As Integer
64
65     For j = 0 To t3
66         tcount(j) = 5
67     Next
68
69     Out(Val("&H3FC"), Val("&H0B"))
70     x = Str(Inp(Val("&H3FE")))
71     x = Str(Inp(Val("&H3FE")))
72     If x = 0 Then '// any button of the remote pressed
73         For i = 0 To t2 '// store 5000 samples
74             Out(Val("&H3FC"), Val("&H0B"))
75             x = Str(Inp(Val("&H3FE")))
76             x = (x And 128)
77             a(i) = x
78         Next
79
80         y1 = a(0)
81         index = 0
82         For i = 1 To t2 '//calculate the length of the signal depending on
83 it's voltage level
84             y2 = a(i)
85             If y2 = y1 Then
86                 counter += 1
87             Else
88                 count(index) = counter
89                 If y1 = 0 Then
90                     tcount(index) = y1 '//store the signal voltage level
91                 Else
92                     tcount(index) = 1
93                 End If
94                 index += 1
95                 counter = 0
96             End If
97             y1 = y2
98         Next
99         '*****
100         For j = 0 To t3
101             count1(j) = count(j)
102         Next

```



```

2   Array.Sort(count1)
3   a1 = count1(35) '//store the shorter length
4   For j = 35 To t3
5       If count1(j) >= a1 * 2 Then
6           a2 = ((count1(j)) + (count1(j + 5))) / 2 '//store the
longer length
7       Exit For
8       End If
9   Next
10
11   '*****
12   '//perform the pulse width protocol on the signal
13   t1 = 0
14   For j = 0 To t3
15       If (tcount(j) = 1 And ((count(j) > (a2 - a1)) And (count(j) <
(a2 + a1)))) Then
16           t1 = t1 + 1
17           tcount2(t1) = 1
18       ElseIf (tcount(j) = 1 And ((count(j) > (a1 - a1)) And (count
(j) < (a1 + a1)))) Then
19           t1 = t1 + 1
20           tcount2(t1) = 0
21       End If
22   Next
23   '*****
24   y1 = tcount2(0)
25   index = 0
26   counter = 0
27   For i = 1 To t1
28       y2 = tcount2(i)
29       If y2 = y1 Then
30           counter += 1
31       Else
32           tcount3(index) = counter
33           index += 1
34           counter = 0
35       End If
36       y1 = y2
37   Next
38   '*****////////////////////////////////////
39   For i = 0 To t3
40
41       str1 = str1 & " " & System.Convert.ToString(count1(i))
42
43   Next
44
45   str1 = ""
46
47   For i = 0 To index
48
49       str1 = str1 & System.Convert.ToString(tcount3(i))
50
51   Next
52   '//store the signal after all previous processing in the aw
variable
53   aw = Val(str1)
54   '//fetesh the signal from txt file to perform the related
operation with it
55   For i = 0 To 28
56
57       If aw = valu(i) Then
58
59           If (i >= 0 And i < 13) Then
60               keybd_event(Sal(i), 0, 0, 0)
61               keybd_event(Sal(i), 0, &H2, 0)
62           End If
63           If i = 13 Or i = 14 Or i = 27 Then
64               keybd_event(&H5B, 0, 0, 0)
65               keybd_event(Sal(i), 0, 0, 0)
66               keybd_event(&H5B, 0, &H2, 0)
67           End If
68       End If
69   Next
70

```



```

32 Array.Sort(count1)
33 a1 = count1(35) '//store the shorter length
34 For j = 35 To t3
35     If count1(j) >= a1 * 2 Then
36         a2 = ((count1(j)) + (count1(j + 5))) / 2 '//store the
37 longer length
38         Exit For
39     End If
40 Next
41
42 *****
43 '//perform the pulse width protocol on the signal
44 t1 = 0
45 For j = 0 To t3
46     If (tcount(j) = 1 And ((count(j) > (a2 - a1)) And (count(j) <
(a2 + a1)))) Then
47         t1 = t1 + 1
48         tcount2(t1) = 1
49     ElseIf (tcount(j) = 1 And ((count(j) > (a1 - a1)) And (count
(j) < (a1 + a1)))) Then
50         t1 = t1 + 1
51         tcount2(t1) = 0
52     End If
53 Next
54 *****
55 y1 = tcount2(0)
56 index = 0
57 counter = 0
58 For i = 1 To t1
59     y2 = tcount2(i)
60     If y2 = y1 Then
61         counter += 1
62     Else
63         tcount3(index) = counter
64         index += 1
65         counter = 0
66     End If
67     y1 = y2
68
69 Next
70 *****////////////////////////////////////
71 For i = 0 To t3
72
73     str1 = str1 & " " & System.Convert.ToString(count1(i))
74
75 Next
76
77 str1 = ""
78
79 For i = 0 To index
80
81     str1 = str1 & System.Convert.ToString(tcount3(i))
82
83 Next
84
85 '//store the signal after all previous processing in the aw
86 variable
87 aw = Val(str1)
88 '//fetesh the signal from txt file to perform the related
89 operation with it
90 For i = 0 To 28
91
92     If aw = valu(i) Then
93
94         If (i >= 0 And i < 13) Then
95             keybd_event(Sal(i), 0, 0, 0)
96             keybd_event(Sal(i), 0, &H2, 0)
97         End If
98         If i = 13 Or i = 14 Or i = 27 Then
99             keybd_event(&H5B, 0, 0, 0)
100             keybd_event(Sal(i), 0, 0, 0)
101             keybd_event(&H5B, 0, &H2, 0)

```



```

keybd_event(Sal(i), 0, &H2, 0)
End If
If i = 15 Or i = 16 Then
    keybd_event(&H12, 0, 0, 0)
    keybd_event(Sal(i), 0, 0, 0)
    keybd_event(&H12, 0, &H2, 0)
    keybd_event(Sal(i), 0, &H2, 0)
End If
If i = 17 Then
    System.Diagnostics.Process.Start("taskmgr.exe")
End If
If i >= 18 And i <= 21 Then
    output = (output Or Val(Sal(i)))
    If (output And &H6) = &H6 Then
        output = (output And &HF9)
        output = (output Or Val(Sal(i)))
        delay()
    End If
    Out(Val("&H378"), output)
End If
If i >= 22 And i <= 25 Then
    output = output And (Not (Val(Sal(i - 4))))
    Out(Val("&H378"), output)
End If
If i = 28 Then
    help()
End If
If i = 26 Then
    End
End If
Exit For
End If

```

Next

```

End If
' /*****

```

Loop

```

End Sub
'//show the help for the program
Public Sub help()

```

```

    If helptest = False Then
        help1.Show()
        helptest = True
    Else
        help1.Hide()
        helptest = False
    End If

```

```

End Sub
'//delay function
Public Sub delay()
    Dim i, dela As Integer
    For i = 1 To 1000000
        dela = dela + 5
        dela = dela - 5
    Dim P As Integer
    For P = 1 To 100
        dela = dela + 5
        dela = dela - 5
    Next

```



```

20     keybd_event(Sal(i), 0, &H2, 0)
21 End If
22 If i = 15 Or i = 16 Then
23     keybd_event(&H12, 0, 0, 0)
24     keybd_event(Sal(i), 0, 0, 0)
25     keybd_event(&H12, 0, &H2, 0)
26     keybd_event(Sal(i), 0, &H2, 0)
27 End If
28 If i = 17 Then
29     System.Diagnostics.Process.Start("taskmgr.exe")
30 End If
31
32 If i >= 18 And i <= 21 Then
33     output = (output Or Val(Sal(i)))
34     If (output And &H6) = &H6 Then
35         output = (output And &HF9)
36         output = (output Or Val(Sal(i)))
37         delay()
38     End If
39     Out(Val("&H378"), output)
40 End If
41 If i >= 22 And i <= 25 Then
42     output = output And (Not (Val(Sal(i - 4))))
43     Out(Val("&H378"), output)
44 End If
45 If i = 28 Then
46     help()
47 End If
48 If i = 26 Then
49     End
50 End If
51 Exit For
52 End If

```

```

53 Next

```

```

54 End If
55 ' /*****

```

```

56 Loop

```

```

57 End Sub
58 '//show the help for the program
59 Public Sub help()

```

```

60     If helptest = False Then
61         help1.Show()
62         helptest = True
63     Else
64         help1.Hide()
65         helptest = False
66     End If

```

```

67 End Sub
68 '//delay function
69 Public Sub delay()
70     Dim i, dela As Integer
71     For i = 1 To 1000000
72         dela = dela + 5
73         dela = dela - 5
74     Dim P As Integer
75     For P = 1 To 100
76         dela = dela + 5
77         dela = dela - 5
78     Next

```


893 Next
894 End Sub
895
896 End Class
897

Appendix B: Datasheets

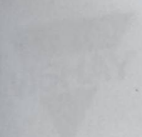


Photo Modules for PCM Remote Control Systems

Available types for different carrier frequencies

TSOP1730	30 kHz	TSOP1735	35 kHz
TSOP1736	36 kHz	TSOP1740	40 kHz
TSOP1738	38 kHz	TSOP1745	45 kHz
TSOP1750	50 kHz		

Description

The TSOP17... series are industrialized receivers for infrared remote control systems. PIN diode and preamplifier are assembled on lead frame, the epoxy package is designed as IR filter. The sophisticated circuit is made of a micro-processed IR remote control receiver series, supporting a range transmission codes.

Appendix B: Datasheets.

Features

- Photo-detector and preamplifier in one package
- Noise filter for PCM frequency
- Improved shielding against electrical field disturbance
- TTL and CMOS compatibility
- Output active low
- Low power consumption
- High immunity against infrared light
- Continuous state transmission possible (up to 2000 m/s)
- Suitable for high speed and low cost

Block Diagram

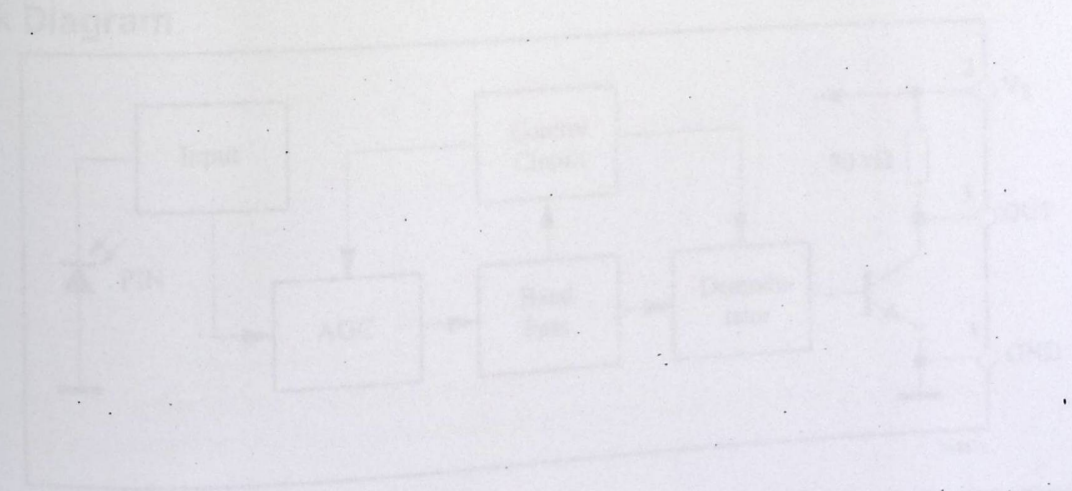


Photo Modules for PCM Remote Control Systems

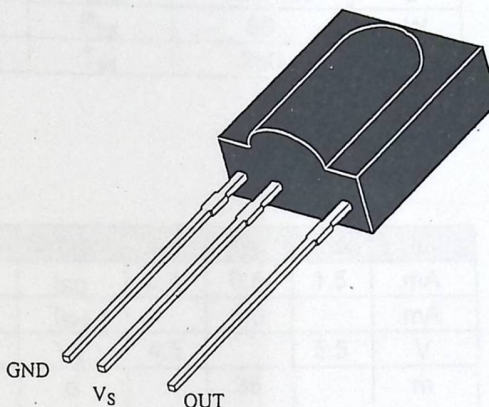
Available types for different carrier frequencies

Type	fo	Type	fo
TSOP1730	30 kHz	TSOP1733	33 kHz
TSOP1736	36 kHz	TSOP1737	36.7 kHz
TSOP1738	38 kHz	TSOP1740	40 kHz
TSOP1756	56 kHz		

Description

The TSOP17.. – series are miniaturized receivers for infrared remote control systems. PIN diode and preamplifier are assembled on lead frame, the epoxy package is designed as IR filter.

The demodulated output signal can directly be decoded by a microprocessor. TSOP17.. is the standard IR remote control receiver series, supporting all major transmission codes.

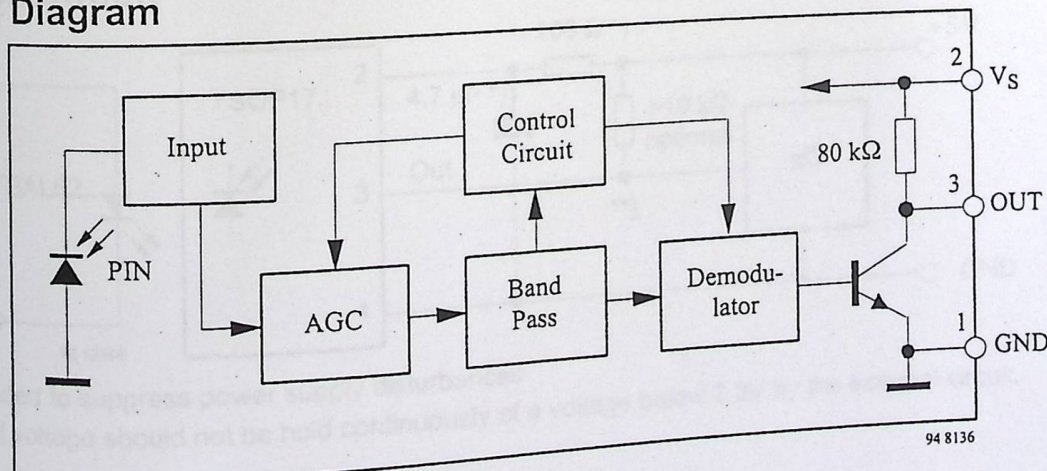


94 8691

Features

- Photo detector and preamplifier in one package
- Internal filter for PCM frequency
- Improved shielding against electrical field disturbance
- TTL and CMOS compatibility
- Output active low
- Low power consumption
- High immunity against ambient light
- Continuous data transmission possible (up to 2400 bps)
- Suitable burst length ≥ 10 cycles/burst

Block Diagram



$T_{amb} = 25^{\circ}\text{C}$

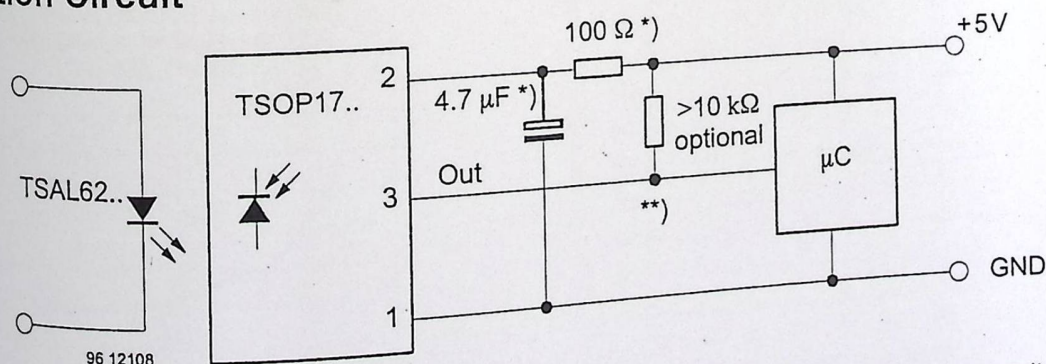
Parameter	Test Conditions	Symbol	Value	Unit
Supply Voltage	(Pin 2)	V_S	-0.3...6.0	V
Supply Current	(Pin 2)	I_S	5	mA
Output Voltage	(Pin 3)	V_O	-0.3...6.0	V
Output Current	(Pin 3)	I_O	5	mA
Junction Temperature		T_j	100	$^{\circ}\text{C}$
Storage Temperature Range		T_{stg}	-25...+85	$^{\circ}\text{C}$
Operating Temperature Range		T_{amb}	-25...+85	$^{\circ}\text{C}$
Power Consumption	($T_{amb} \leq 85^{\circ}\text{C}$)	P_{tot}	50	mW
Soldering Temperature	$t \leq 10\text{ s}$, 1 mm from case	T_{sd}	260	$^{\circ}\text{C}$

Basic Characteristics

$T_{amb} = 25^{\circ}\text{C}$

Parameter	Test Conditions	Symbol	Min	Typ	Max	Unit
Supply Current (Pin 2)	$V_S = 5\text{ V}$, $E_V = 0$	I_{SD}	0.4	0.6	1.5	mA
	$V_S = 5\text{ V}$, $E_V = 40\text{ klx}$, sunlight	I_{SH}		1.0		mA
Supply Voltage (Pin 2)		V_S	4.5		5.5	V
Transmission Distance	$E_V = 0$, test signal see fig.7, IR diode TSAL6200, $I_F = 400\text{ mA}$	d		35		m
Output Voltage Low (Pin 3)	$I_{OSL} = 0.5\text{ mA}$, $E_e = 0.7\text{ mW/m}^2$, $f = f_o$, $t_p/T = 0.4$	V_{OSL}			250	mV
Irradiance (30 – 40 kHz)	Pulse width tolerance: $t_{pi} - 5/f_o < t_{po} < t_{pi} + 6/f_o$, test signal (see fig.7)	$E_{e\text{ min}}$		0.35	0.5	mW/m^2
Irradiance (56 kHz)	Pulse width tolerance: $t_{pi} - 5/f_o < t_{po} < t_{pi} + 6/f_o$, test signal (see fig.7)	$E_{e\text{ min}}$		0.4	0.6	mW/m^2
Irradiance	$t_{pi} - 5/f_o < t_{po} < t_{pi} + 6/f_o$	$E_{e\text{ max}}$	30			W/m^2
Directivity	Angle of half transmission distance	$\Phi_{1/2}$		± 45		deg

Application Circuit



*) recommended to suppress power supply disturbances
**) The output voltage should not be hold continuously at a voltage below 3.3V by the external circuit.

Suitable Data Format

The circuit of the TSOP17.. is designed in that way that unexpected output pulses due to noise or disturbance signals are avoided. A bandpassfilter, an integrator stage and an automatic gain control are used to suppress such disturbances.

The distinguishing mark between data signal and disturbance signal are carrier frequency, burst length and duty cycle.

The data signal should fullfill the following condition:

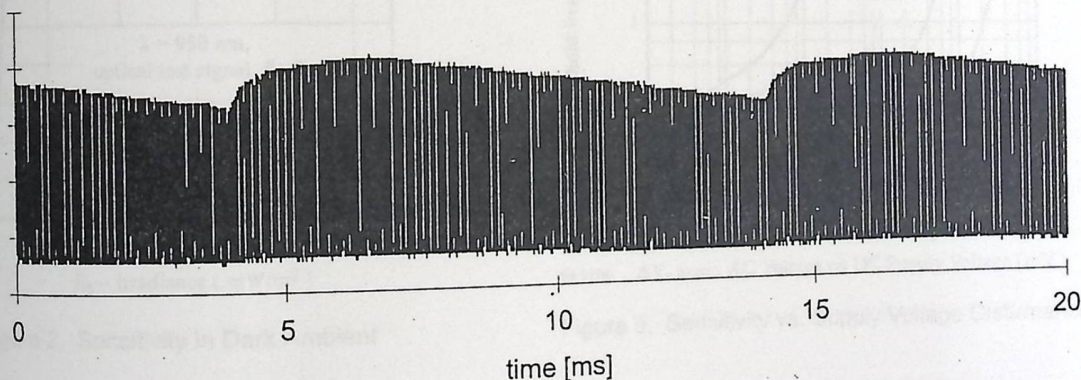
- Carrier frequency should be close to center frequency of the bandpass (e.g. 38kHz).
- Burst length should be 10 cycles/burst or longer.
- After each burst which is between 10 cycles and 70 cycles a gap time of at least 14 cycles is necessary.
- For each burst which is longer than 1.8ms a corresponding gap time is necessary at some time in the data stream. This gap time should have at least same length as the burst.
- Up to 1400 short bursts per second can be received continuously.

Some examples for suitable data format are: NEC Code, Toshiba Micom Format, Sharp Code, RC5 Code, RC6 Code, R-2000 Code, Sony Format (SIRCS).

When a disturbance signal is applied to the TSOP17.. it can still receive the data signal. However the sensitivity is reduced to that level that no unexpected pulses will occur.

Some examples for such disturbance signals which are suppressed by the TSOP17.. are:

- DC light (e.g. from tungsten bulb or sunlight)
- Continuous signal at 38kHz or at any other frequency
- Signals from fluorescent lamps with electronic ballast (an example of the signal modulation is in the figure below).



IR Signal from Fluorescent Lamp with low Modulation

Typical Characteristics ($T_{amb} = 25^{\circ}\text{C}$ unless otherwise specified)

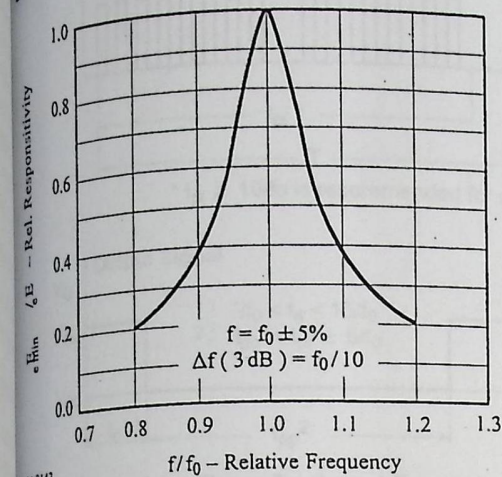


Figure 1. Frequency Dependence of Responsivity

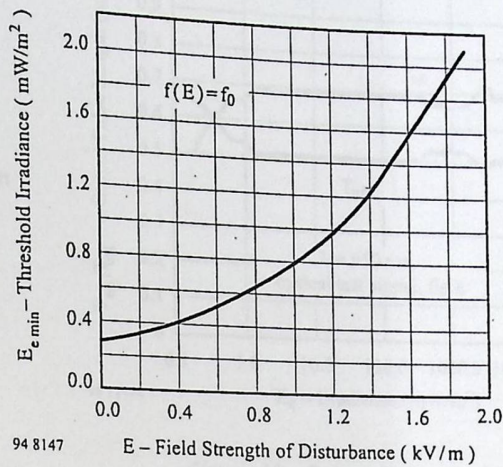


Figure 4. Sensitivity vs. Electric Field Disturbances

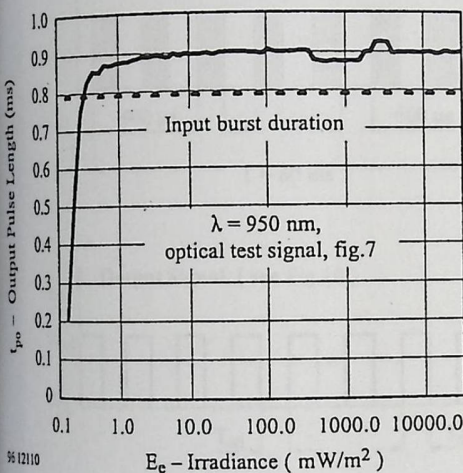


Figure 2. Sensitivity in Dark Ambient

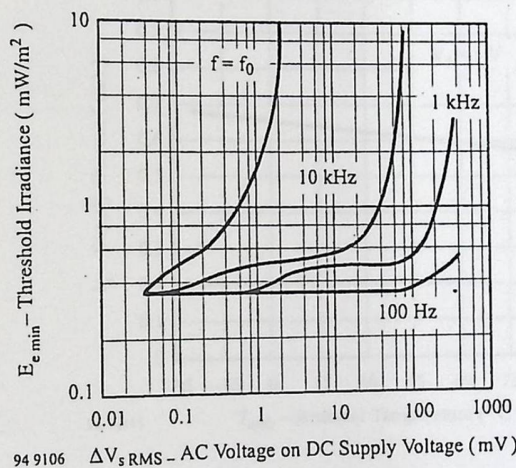


Figure 5. Sensitivity vs. Supply Voltage Disturbances

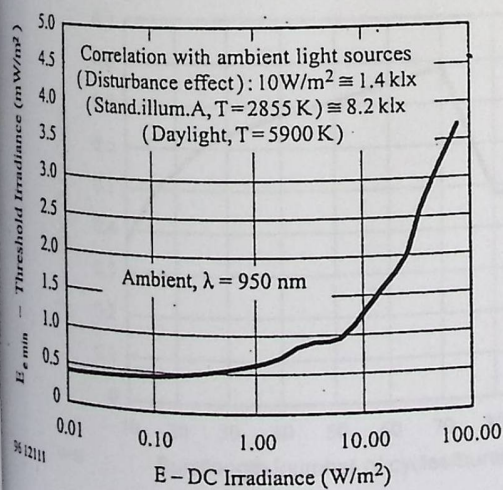


Figure 3. Sensitivity in Bright Ambient

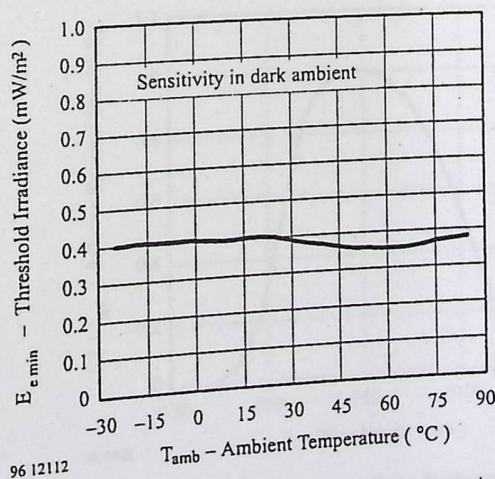


Figure 6. Sensitivity vs. Ambient Temperature

Typical Characteristics ($T_{amb} = 25^{\circ}\text{C}$ unless otherwise specified)

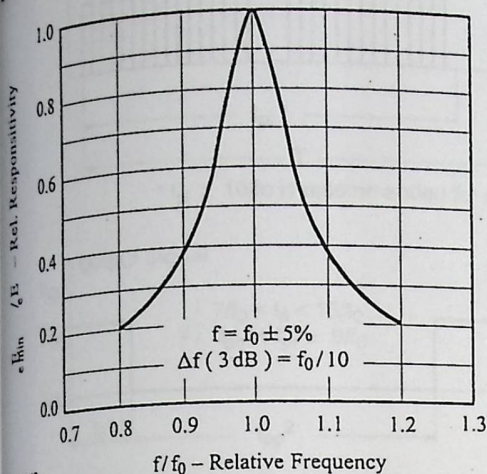


Figure 1. Frequency Dependence of Responsivity

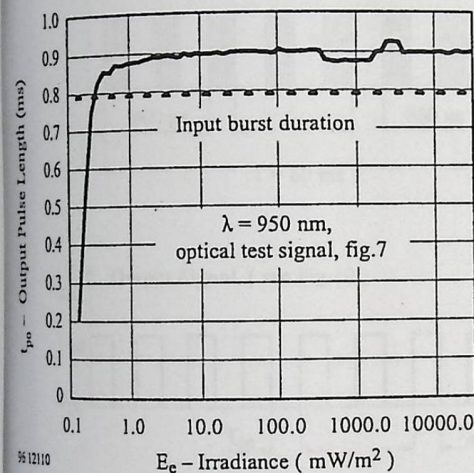


Figure 2. Sensitivity in Dark Ambient

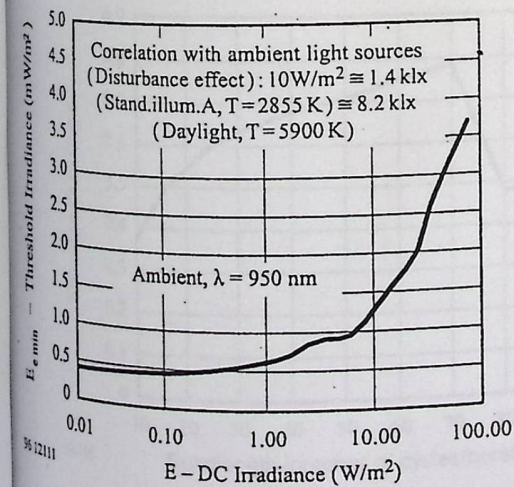


Figure 3. Sensitivity in Bright Ambient

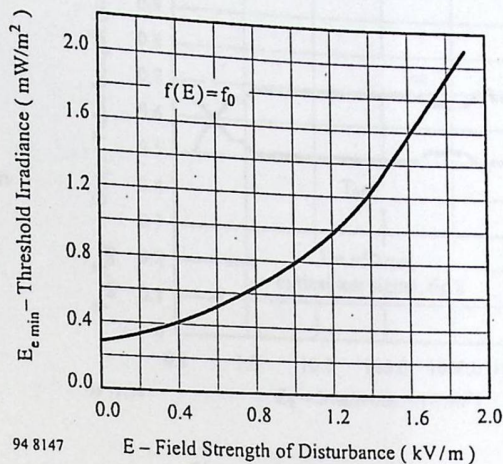


Figure 4. Sensitivity vs. Electric Field Disturbances

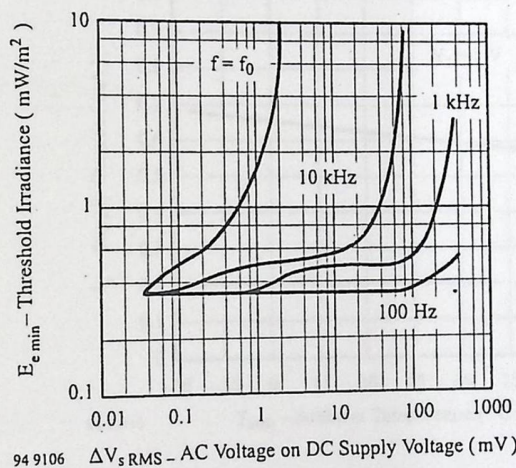


Figure 5. Sensitivity vs. Supply Voltage Disturbances

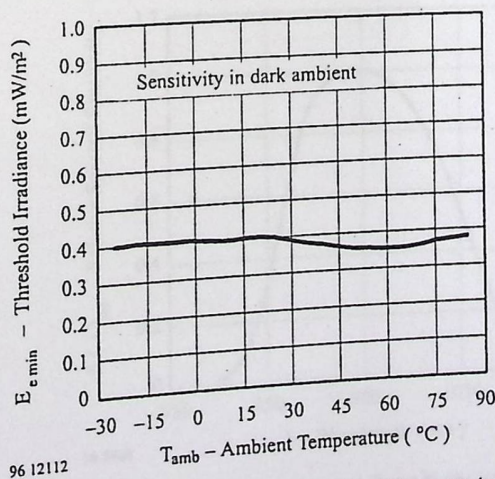


Figure 6. Sensitivity vs. Ambient Temperature

Typical Characteristics ($T_{amb} = 25^{\circ}\text{C}$ unless otherwise specified)

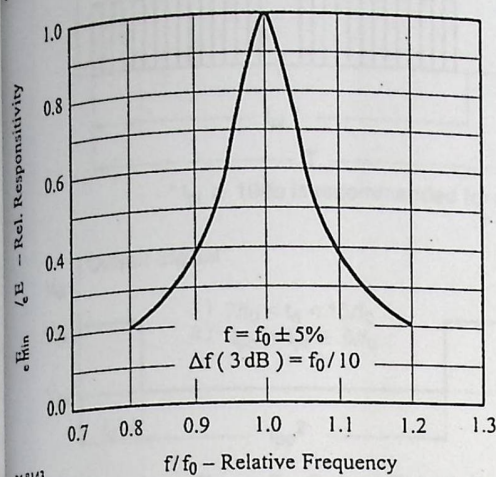


Figure 1. Frequency Dependence of Responsivity

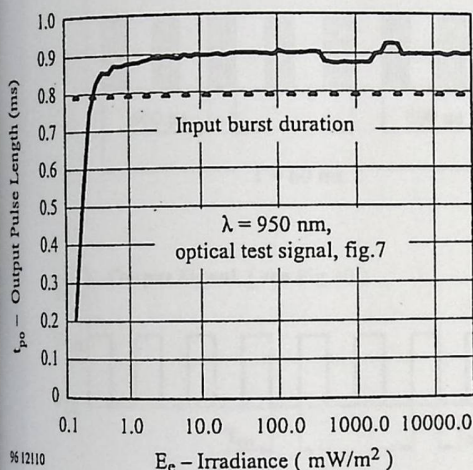


Figure 2. Sensitivity in Dark Ambient

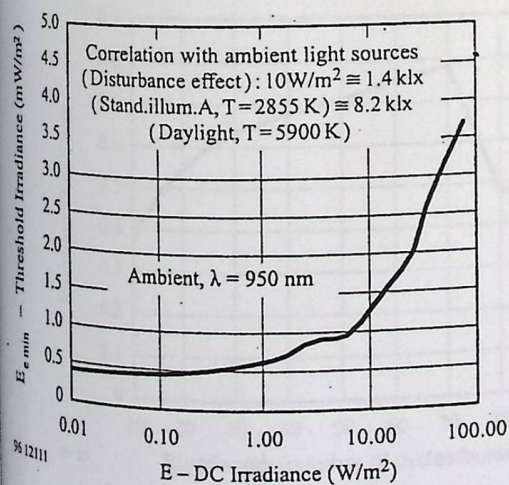


Figure 3. Sensitivity in Bright Ambient

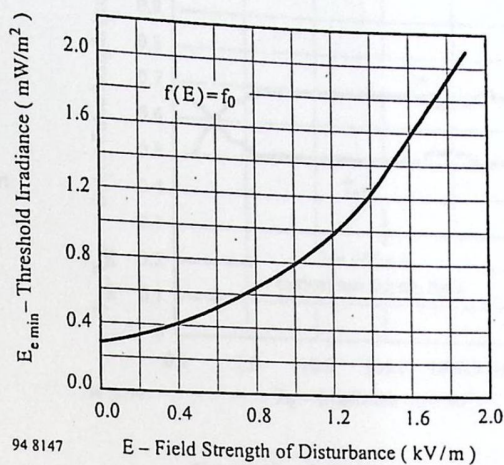


Figure 4. Sensitivity vs. Electric Field Disturbances

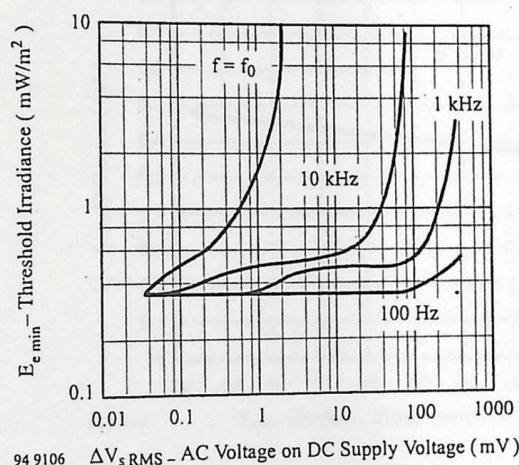


Figure 5. Sensitivity vs. Supply Voltage Disturbances

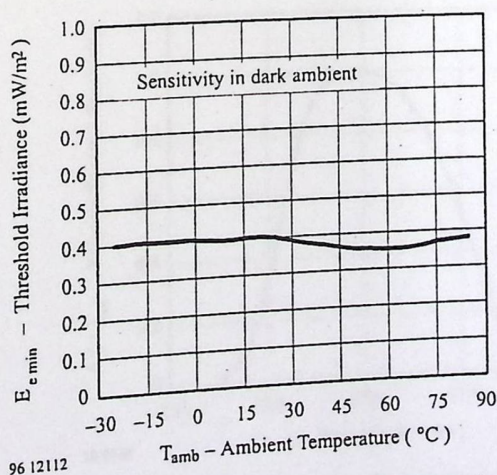


Figure 6. Sensitivity vs. Ambient Temperature

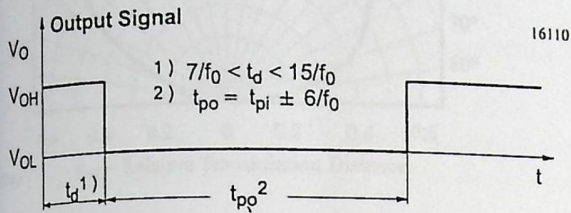
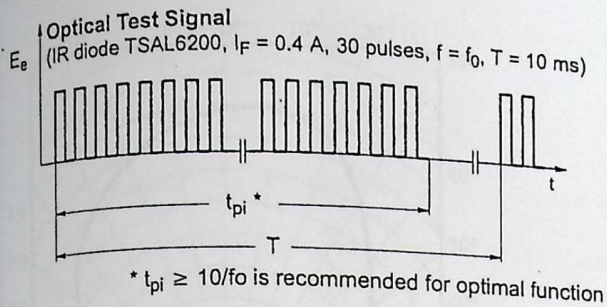


Figure 7. Output Function

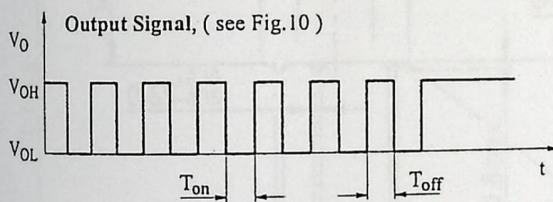
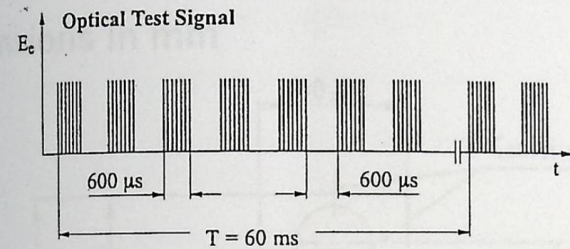


Figure 8. Output Function

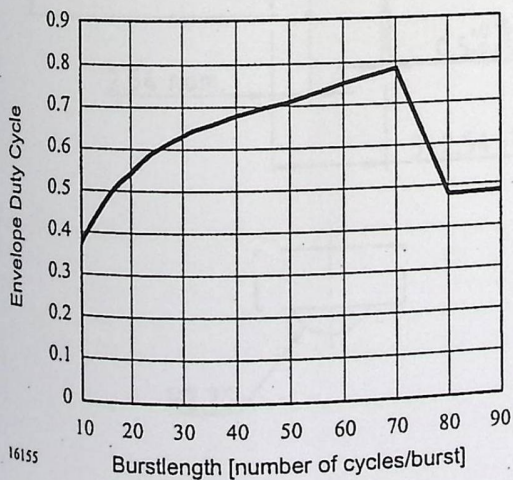


Figure 9. Max. Envelope Duty Cycle vs. Burstlength

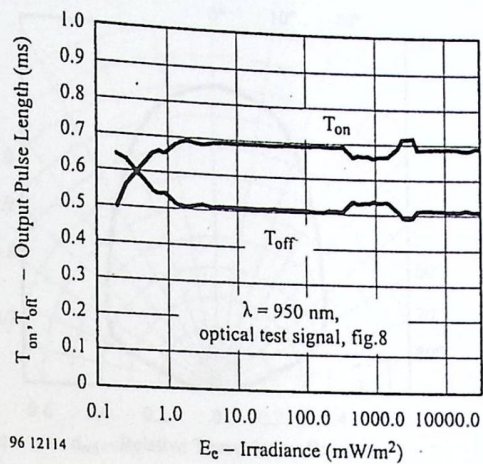


Figure 10. Output Pulse Diagram

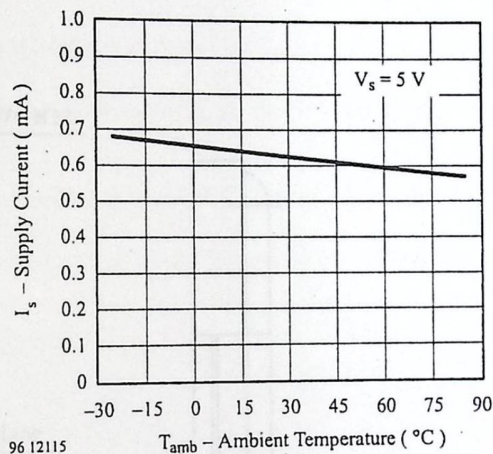


Figure 11. Supply Current vs. Ambient Temperature

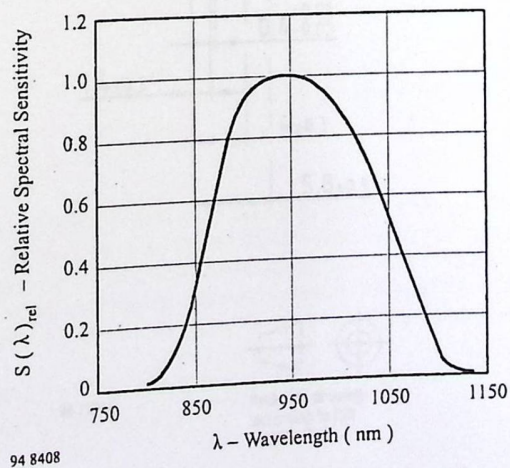
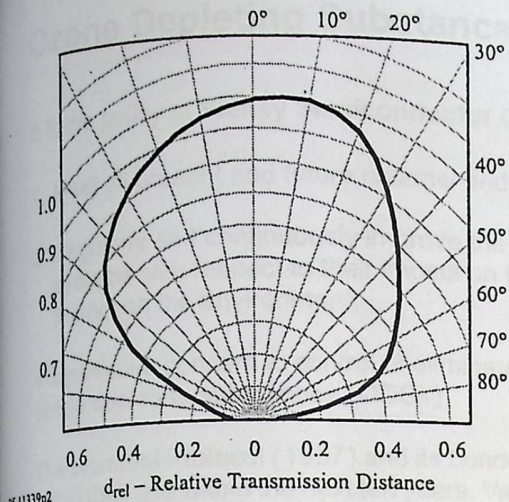
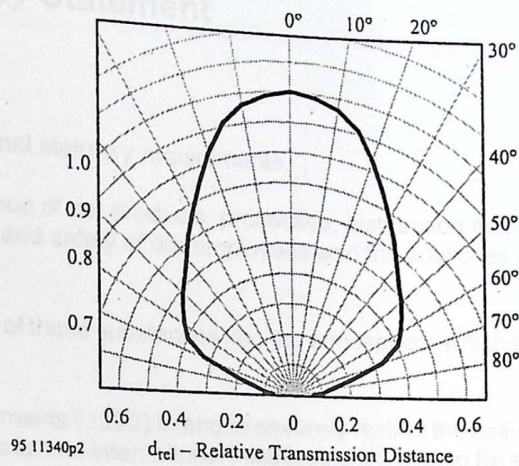
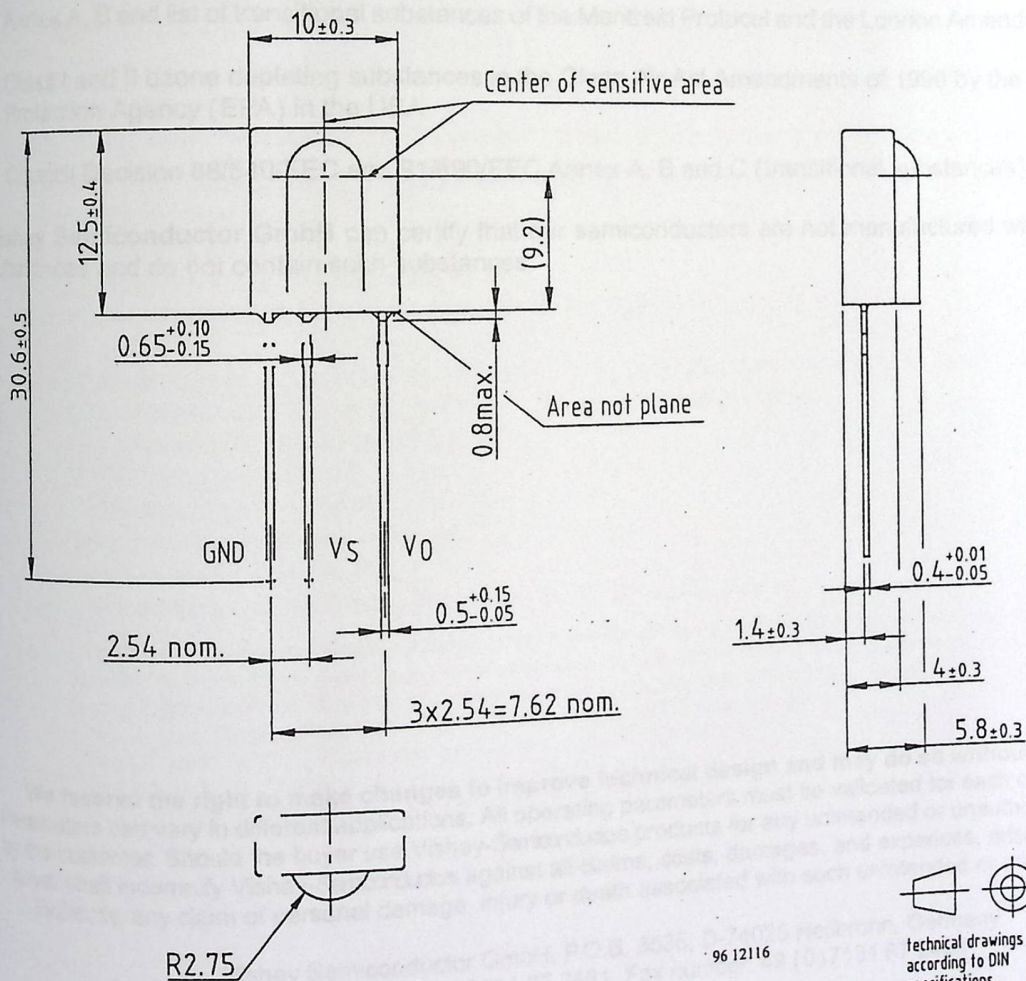


Figure 12. Relative Spectral Sensitivity vs. Wavelength

Figure 13. Vertical Directivity ϕ_y Figure 14. Horizontal Directivity ϕ_x

Dimensions in mm



technical drawings
according to DIN
specifications



Ozone Depleting Substances Policy Statement

It is the policy of Vishay Semiconductor GmbH to

1. Meet all present and future national and international statutory requirements.
2. Regularly and continuously improve the performance of our products, processes, distribution and operating systems with respect to their impact on the health and safety of our employees and the public, as well as their impact on the environment.

It is particular concern to control or eliminate releases of those substances into the atmosphere which are known as ozone depleting substances (ODSs).

The Montreal Protocol (1987) and its London Amendments (1990) intend to severely restrict the use of ODSs and forbid their use within the next ten years. Various national and international initiatives are pressing for an earlier ban on these substances.

Vishay Semiconductor GmbH has been able to use its policy of continuous improvements to eliminate the use of ODSs listed in the following documents.

1. Annex A, B and list of transitional substances of the Montreal Protocol and the London Amendments respectively
2. Class I and II ozone depleting substances in the Clean Air Act Amendments of 1990 by the Environmental Protection Agency (EPA) in the USA
3. Council Decision 88/540/EEC and 91/690/EEC Annex A, B and C (transitional substances) respectively.

Vishay Semiconductor GmbH can certify that our semiconductors are not manufactured with ozone depleting substances and do not contain such substances.

We reserve the right to make changes to improve technical design and may do so without further notice.
Parameters can vary in different applications. All operating parameters must be validated for each customer application by the customer. Should the buyer use Vishay-Semiconductors products for any unintended or unauthorized application, the buyer shall indemnify Vishay-Semiconductors against all claims, costs, damages, and expenses, arising out of, directly or indirectly, any claim of personal damage, injury or death associated with such unintended or unauthorized use.

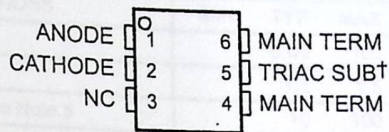
Vishay Semiconductor GmbH, P.O.B. 3535, D-74025 Heilbronn, Germany
Telephone: 49 (0)7131 67 2831, Fax number: 49 (0)7131 67 2423

MOC3020 THRU MOC3023 OPTOCOUPLED/OPTOISOLATORS

SOES025 - OCTOBER 1986 - REVISED OCTOBER 1995

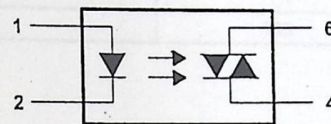
- 250 V Phototriac Driver Output
- Gallium-Arsenide-Diode Infrared Source and Optically-Coupled Silicon Triac Driver (Bilateral Switch)
- UL Recognized ... File Number E65085
- High Isolation ... 7500 V Peak
- Output Driver Designed for 220 V ac
- Standard 6-Terminal Plastic DIP
- Directly Interchangeable with Motorola MOC3020, MOC3021, MOC3022, and MOC3023
- Direct Replacements for:
 - TRW Opton OPI3020, OPI3021, OPI3022, and OPI3023;
 - General Instrument MCP3020, MCP3021, and MCP3022;
 - General Electric GE3020, GE3021, GE3022, and GE3023

MOC3020 - MOC3023 ... PACKAGE
(TOP VIEW)



† Do not connect this terminal
NC - No internal connection

logic diagram



absolute maximum ratings at 25°C free-air temperature (unless otherwise noted)†

Input-to-output peak voltage, 5 s maximum duration, 60 Hz (see Note 1)	7.5 kV
Input diode reverse voltage	3 V
Input diode forward current, continuous	50 mA
Output repetitive peak off-state voltage	400 V
Output on-state current, total rms value (50-60 Hz, full sine wave): $T_A = 25^\circ\text{C}$	100 mA
$T_A = 70^\circ\text{C}$	50 mA
Output driver nonrepetitive peak on-state current ($t_w = 10$ ms, duty cycle = 10%, see Figure 7)	1.2 A
Continuous power dissipation at (or below) 25°C free-air temperature:	100 mW
Infrared-emitting diode (see Note 2)	300 mW
Phototriac (see Note 3)	330 mW
Total device (see Note 4)	-40°C to 100°C
Operating junction temperature range, T_J	-40°C to 150°C
Storage temperature range, T_{stg}	260°C
Lead temperature 1,6 (1/16 inch) from case for 10 seconds	

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

- NOTES:
1. Input-to-output peak voltage is the internal device dielectric breakdown rating.
 2. Derate linearly to 100°C free-air temperature at the rate of 1.33 mW/°C.
 3. Derate linearly to 100°C free-air temperature at the rate of 4 mW/°C.
 4. Derate linearly to 100°C free-air temperature at the rate of 4.4 mW/°C.

Copyright © 1995, Texas Instruments Incorporated

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

**TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

MOC3020 THRU MOC3023 OPTOCOUPERS/OPTOISOLATORS

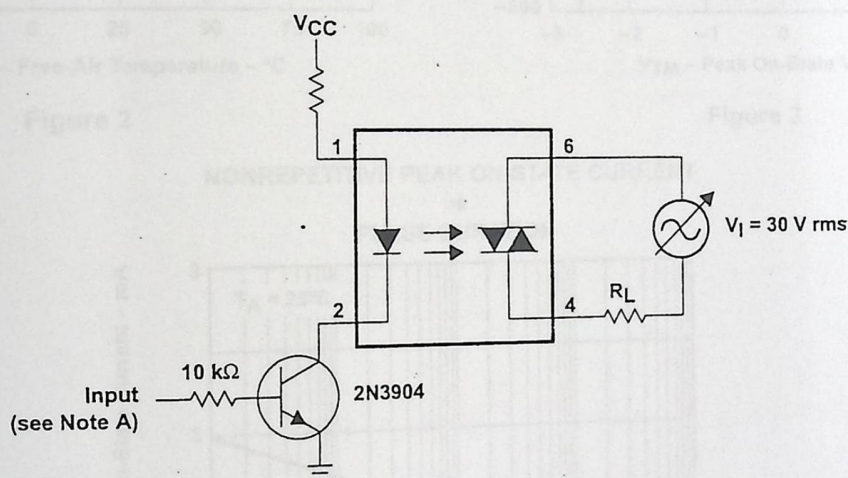
SDS025 - OCTOBER 1986 - REVISED OCTOBER 1995

electrical characteristics at 25°C free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
I_R	Static reverse current	$V_R = 3\text{ V}$				
V_F	Static forward voltage	$I_F = 10\text{ mA}$		0.05	100	μA
$I_{(DRM)}$	Repetitive off-state current, either direction	$V_{(DRM)} = 400\text{ V}$, See Note 5		1.2	1.5	V
dv/dt	Critical rate of rise of off-state voltage	See Figure 1		10	100	nA
$dv/dt(c)$	Critical rate of rise of commutating voltage	$I_O = 15\text{ mA}$, See Figure 1		100		V/ μs
I_{FT}	Input trigger current, either direction	MOC3020		0.15		V/ μs
		MOC3021		15	30	mA
		MOC3022		8	15	
		MOC3023		5	10	
V_{TM}	Peak on-state voltage, either direction	$I_{TM} = 100\text{ mA}$		3	5	V
I_H	Holding current, either direction			1.4	3	V
				100		μA

NOTE 5: Test voltage must be applied at a rate no higher than 12 V/ μs .

PARAMETER MEASUREMENT INFORMATION



NOTE A. The critical rate of rise of off-state voltage, dv/dt , is measured with the input at 0 V. The frequency of V_{in} is increased until the phototriac turns on. This frequency is then used to calculate the dv/dt according to the formula:

$$dv/dt = 2 \sqrt{2\pi f V_{in}}$$

The critical rate of rise of commutating voltage, $dv/dt(c)$, is measured by applying occasional 5-V pulses to the input and increasing the frequency of V_{in} until the phototriac stays on (latches) after the input pulse has ceased. With no further input pulses, the frequency of V_{in} is then gradually decreased until the phototriac turns off. The frequency at which turn-off occurs may then be used to calculate the $dv/dt(c)$ according to the formula shown above.

Figure 1. Critical Rate of Rise Test Circuit

**TEXAS
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

TYPICAL CHARACTERISTICS

EMITTING-DIODE TRIGGER CURRENT (NORMALIZED)
vs
FREE-AIR TEMPERATURE

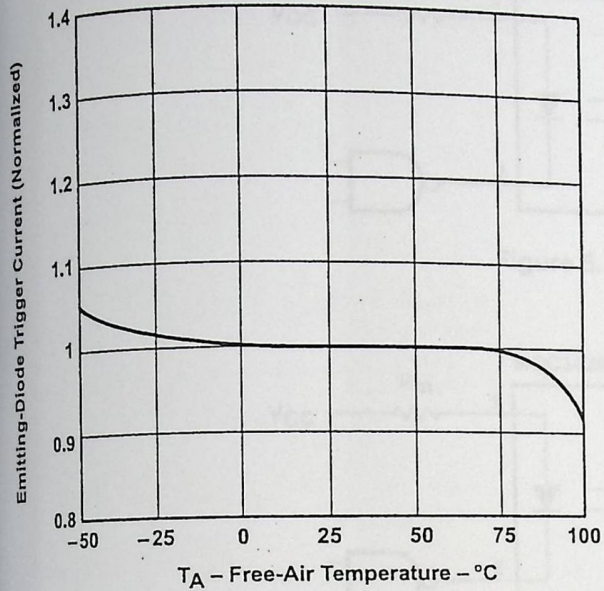


Figure 2

ON-STATE CHARACTERISTICS

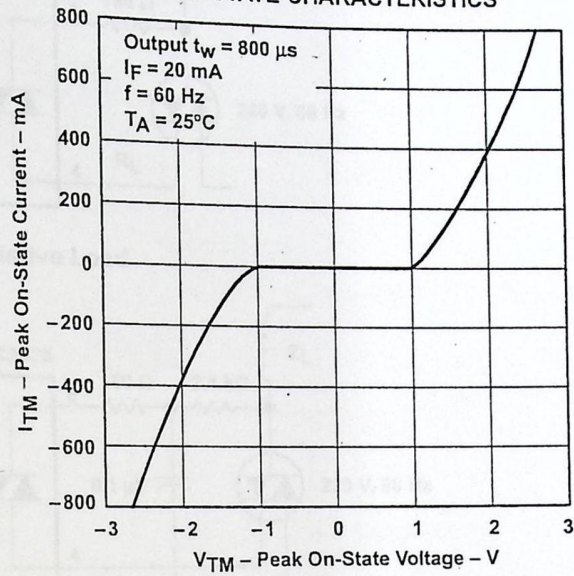


Figure 3

NONREPETITIVE PEAK ON-STATE CURRENT vs PULSE DURATION

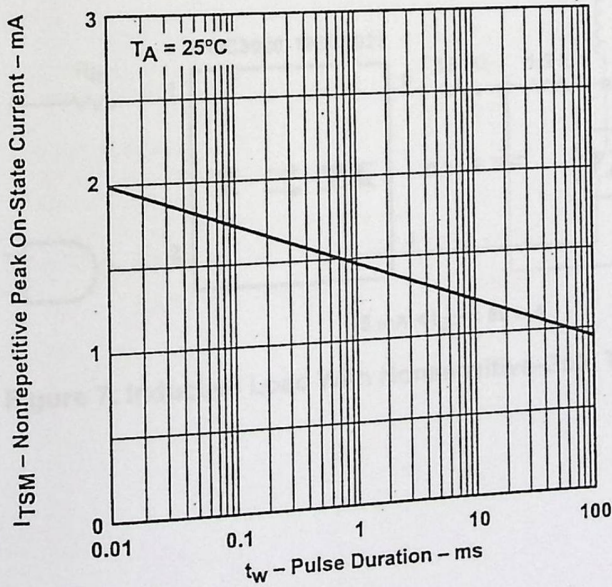


Figure 4

APPLICATIONS INFORMATION

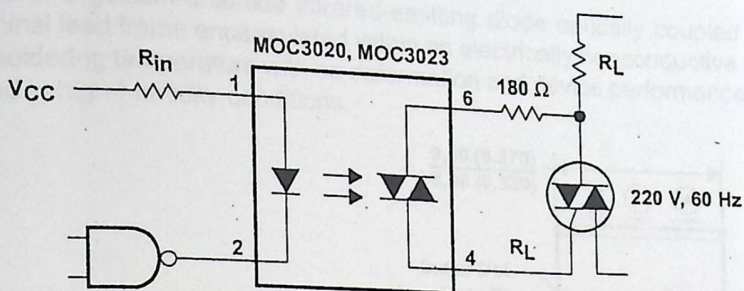


Figure 5. Resistive Load

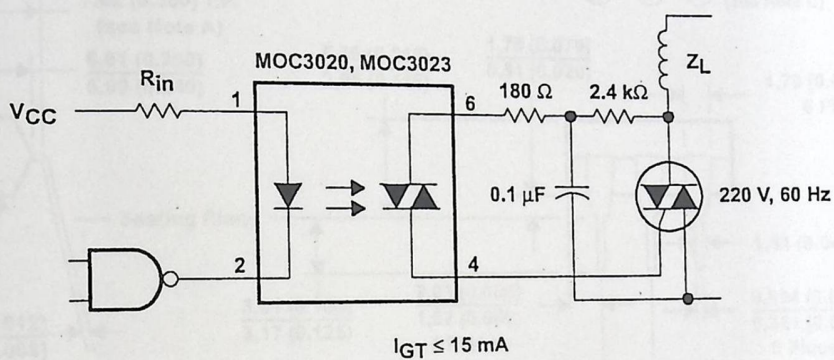


Figure 6. Inductive Load With Sensitive-Gate Triac

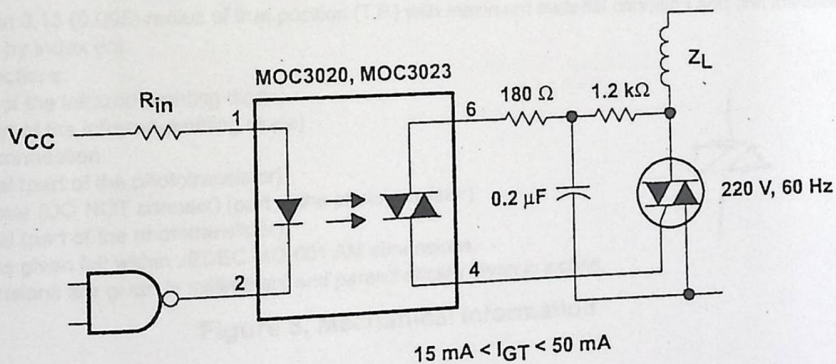


Figure 7. Inductive Load With Nonsensitive-Gate Triac

APPLICATIONS INFORMATION

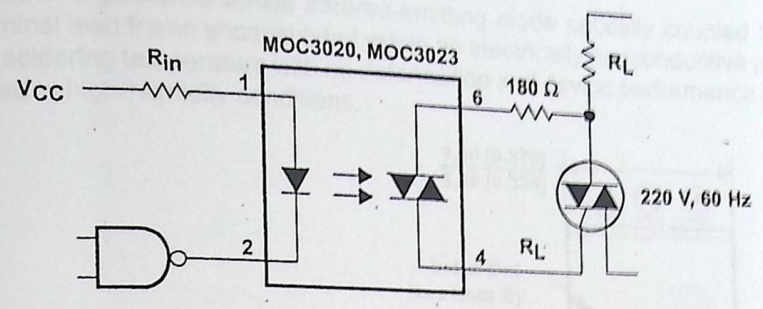


Figure 5. Resistive Load

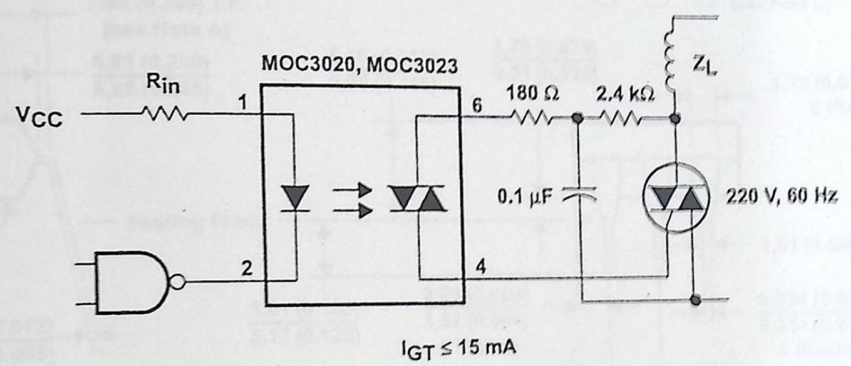


Figure 6. Inductive Load With Sensitive-Gate Triac

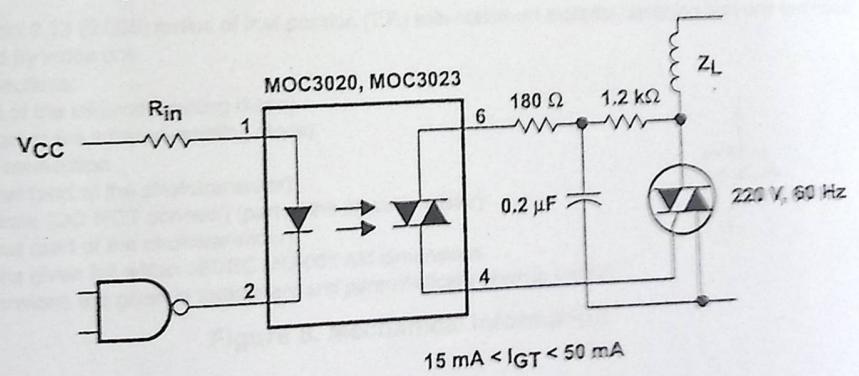


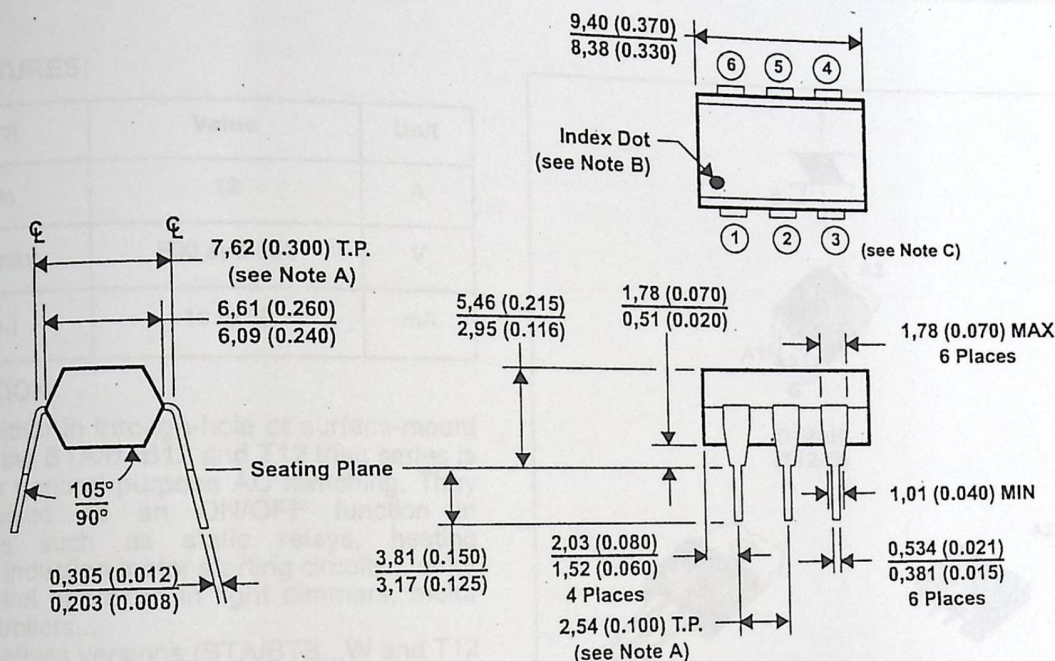
Figure 7. Inductive Load With Nonsensitive-Gate Triac

MOC3020 THRU MOC3023 OPTOCOUPLEDERS/OPTOISOLATORS

SOES025 - OCTOBER 1986 - REVISED OCTOBER 1995

MECHANICAL INFORMATION

Each device consists of a gallium-arsenide infrared-emitting diode optically coupled to a silicon phototriac mounted on a 6-terminal lead frame encapsulated within an electrically nonconductive plastic compound. The case can withstand soldering temperature with no deformation and device performance characteristics remain stable when operated in high-humidity conditions.



NOTES: A. Leads are within 0.13 (0.005) radius of true position (T.P.) with maximum material condition and unit installed.

B. Pin 1 identified by index dot.

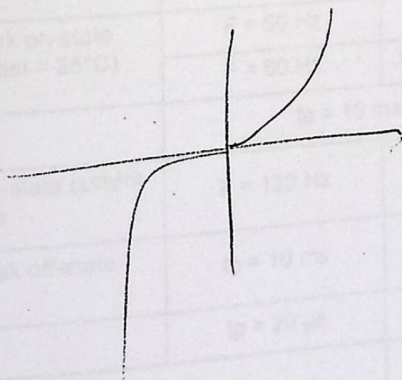
C. Terminal connections:

1. Anode (part of the infrared-emitting diode)
2. Cathode (part of the infrared-emitting diode)
3. No internal connection
4. Main terminal (part of the phototransistor)
5. Triac Substrate (DO NOT connect) (part of the phototransistor)
6. Main terminal (part of the phototransistor)

D. The dimensions given fall within JEDEC MO-001 AM dimensions.

E. All linear dimensions are given in millimeters and parenthetically given in inches.

Figure 8. Mechanical Information





BTA/BTB12 and T12 Series

SNUBBERLESS™, LOGIC LEVEL & STANDARD

12A TRIACS

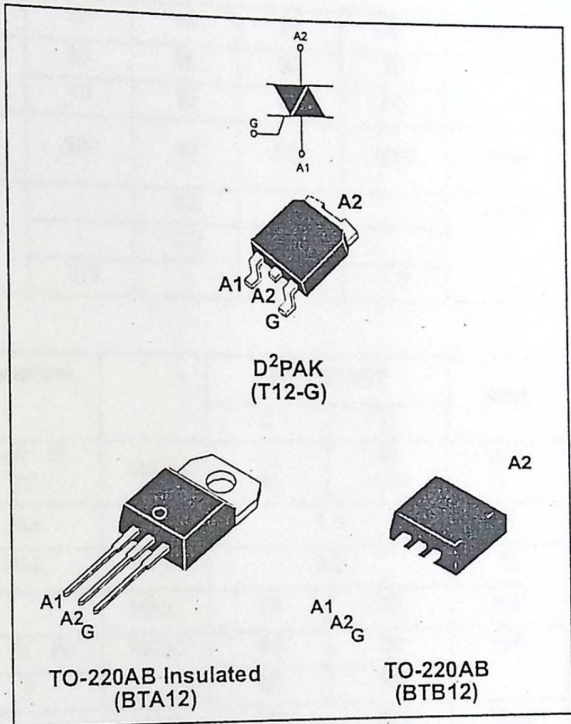
MAIN FEATURES:

Symbol	Value	Unit
$I_{T(RMS)}$	12	A
V_{DRM}/V_{RRM}	600 and 800	V
$I_{GT}(Q_1)$	10 to 50	mA

DESCRIPTION

Available either in through-hole or surface-mount packages, the BTA/BTB12 and T12 triac series is suitable for general purpose AC switching. They can be used as an ON/OFF function in applications such as static relays, heating regulation, induction motor starting circuits... or for phase control operation in light dimmers, motor speed controllers,...

The snubberless versions (BTA/BTB...W and T12 series) are specially recommended for use on inductive loads, thanks to their high commutation performances. By using an internal ceramic pad, the BTA series provides voltage insulated tab (rated at 2500V RMS) complying with UL standards (File ref.: E81734)



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter			Value	Unit
$I_{T(RMS)}$	RMS on-state current (full sine wave)	D²PAK/TO-220AB	$T_c = 105^{\circ}C$	12	A
		TO-220AB Ins.	$T_c = 90^{\circ}C$		
I_{TSM}	Non repetitive surge peak on-state current (full cycle, T_j initial = $25^{\circ}C$)	F = 50 Hz	t = 20 ms	120	A
		F = 60 Hz	t = 16.7 ms	126	
I_t^2	I_t^2 Value for fusing	tp = 10 ms		100	A^2s
di/dt	Critical rate of rise of on-state current $I_G = 2 \times I_{GT}$, tr ≤ 100 ns	F = 120 Hz	$T_j = 125^{\circ}C$	50	A/μs
V_{DSM}/V_{RSM}	Non repetitive surge peak off-state voltage	tp = 10 ms	$T_j = 25^{\circ}C$	$V_{DRM}/V_{RRM} + 100$	V
I_{GM}		tp = 20 μs	$T_j = 125^{\circ}C$	4	A
$P_{G(AV)}$	Average gate power dissipation	$T_j = 125^{\circ}C$		1	W
T_{stg} T_j				- 40 to + 150 - 40 to + 125	$^{\circ}C$

BTA/BTB12 and T12 Series

ELECTRICAL CHARACTERISTICS ($T_J = 25^\circ\text{C}$, unless otherwise specified)

■ SNUBBERLESS™ and LOGIC LEVEL (3 Quadrants)

Symbol	Test Conditions	Quadrant		T12	BTA/BTB12			Unit
				T1235	SW	CW	BW	
$I_{GT}(1)$	$V_D = 12\text{ V}$ $R_L = 30\ \Omega$	I - II - III	MAX.	35	10	35	50	mA
V_{GT}		I - II - III	MAX.					
V_{GD}	$V_D = V_{DRM}$ $R_L = 3.3\text{ k}\Omega$ $T_J = 125^\circ\text{C}$	I - II - III	MIN.		1.3			V
					0.2			V
$I_H(2)$	$I_T = 100\text{ mA}$		MAX.	35	15	35	50	mA
I_L	$I_G = 1.2 I_{GT}$	I - III	MAX.	50	25	50	70	mA
		II		60	30	60	80	mA
$dV/dt(2)$	$V_D = 67\% V_{DRM}$ gate open $T_J = 125^\circ\text{C}$		MIN.	500	40	500	1000	V/ μs
$(dI/dt)c(2)$	$(dV/dt)c = 0.1\text{ V}/\mu\text{s}$ $T_J = 125^\circ\text{C}$		MIN.	-	6.5	-	-	A/ms
	$(dV/dt)c = 10\text{ V}/\mu\text{s}$ $T_J = 125^\circ\text{C}$			-	2.9	-	-	
	Without snubber $T_J = 125^\circ\text{C}$			6.5	-	6.5	12	

■ STANDARD (4 Quadrants)

Symbol	Test Conditions	Quadrant		BTA/BTB06		Unit
				C	B	
$I_{GT}(1)$	$V_D = 12\text{ V}$ $R_L = 30\ \Omega$	I - II - III IV	MAX.	25 50	50 100	mA
V_{GT}		ALL	MAX.	1.3		V
V_{GD}	$V_D = V_{DRM}$ $R_L = 3.3\text{ k}\Omega$ $T_J = 125^\circ\text{C}$	ALL	MIN.	0.2		V
$I_H(2)$	$I_T = 500\text{ mA}$		MAX.	25	50	mA
I_L	$I_G = 1.2 I_{GT}$	I - III - IV	MAX.	40	50	mA
		II		80	100	mA
$dV/dt(2)$	$V_D = 67\% V_{DRM}$ gate open $T_J = 125^\circ\text{C}$		MIN.	200	400	V/ μs
$(dI/dt)c(2)$	$(dI/dt)c = 5.3\text{ A/ms}$ $T_J = 125^\circ\text{C}$		MIN.	5	10	V/ μs

STATIC CHARACTERISTICS

Symbol	Test Conditions			Value	Unit
$V_T(2)$	$I_{TM} = 17\text{ A}$ $t_p = 380\ \mu\text{s}$	$T_J = 25^\circ\text{C}$	MAX.	1.55	V
$V_{th}(2)$	Threshold voltage	$T_J = 125^\circ\text{C}$	MAX.	0.85	V
$R_d(2)$	Dynamic resistance	$T_J = 125^\circ\text{C}$	MAX.	35	m Ω
I_{DRM} I_{RRM}	$V_{DRM} = V_{RRM}$	$T_J = 25^\circ\text{C}$	MAX.	5	μA
		$T_J = 125^\circ\text{C}$		1	mA

Note 1: minimum I_{GT} is guaranteed at 5% of I_{GT} max.

Note 2: for both polarities of A2 referenced to A1



BTA/BTB12 and T12 Series

THERMAL RESISTANCES

Symbol	Parameter		Value	Unit
$R_{th(j-c)}$	Junction to case (AC)		D ² PAK/TO-220AB	°C/W
			TO-220AB Insulated	
$R_{th(j-a)}$	Junction to ambient	S = 1 cm ²	D ² PAK	°C/W
			TO-220AB	
			TO-220AB Insulated	

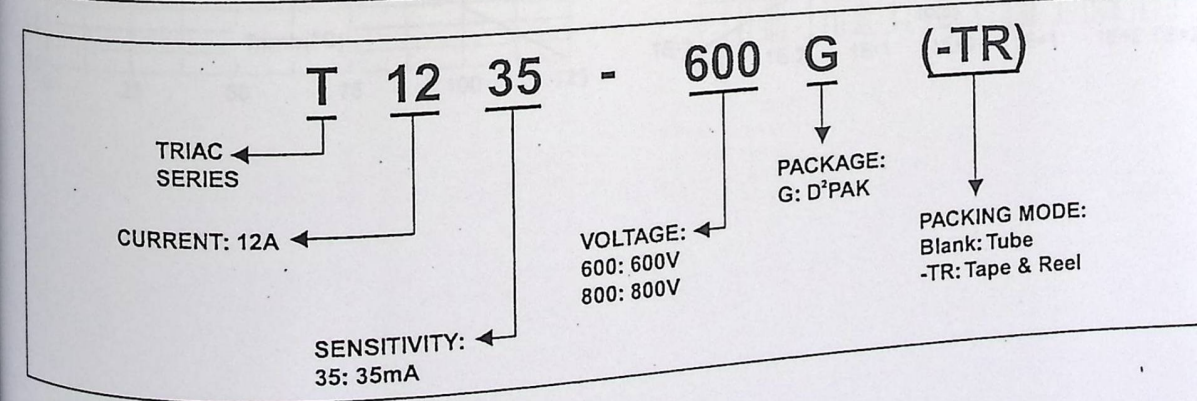
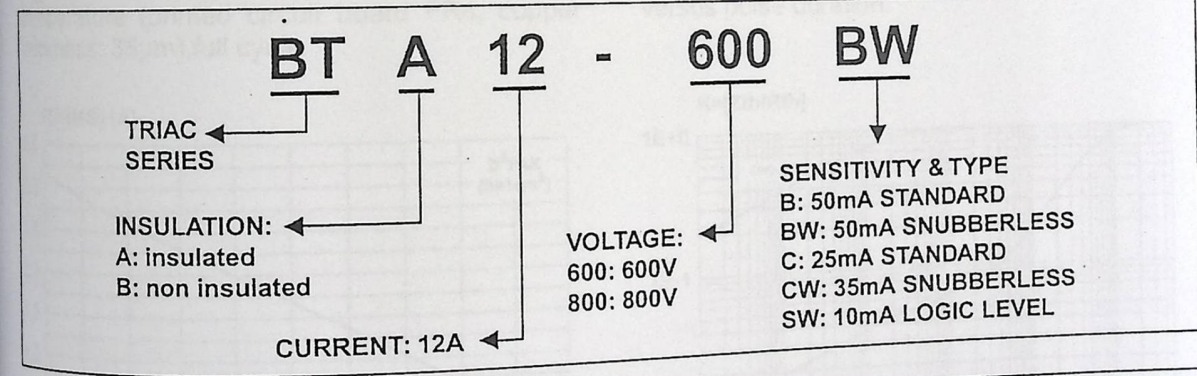
S = Copper surface under tab

PRODUCT SELECTOR

Part Number	Voltage (xxx)		Sensitivity	Type	Package
	600 V	800 V			
BTA/BTB12-xxxB	X	X	50 mA	Standard	TO-220AB
BTA/BTB12-xxxBW	X	X	50 mA	Snubberless	TO-220AB
BTA/BTB12-xxxC	X	X	25 mA	Standard	TO-220AB
BTA/BTB12-xxxCW	X	X	35 mA	Snubberless	TO-220AB
BTA/BTB12-xxxSW	X	X	10 mA	Logic level	TO-220AB
T1235-xxxG	X	X	35 mA	Snubberless	D ² PAK

BTB: non insulated TO-220AB package

ORDERING INFORMATION



OTHER INFORMATION

Part Number	Marking	Weight	Base quantity	Packing mode
BTA/BTB12-xxxzy	BTA/BTB12-xxxzy	2.3 g	250	Bulk
T1235-xxxG	T1235xxxG	1.5 g	50	Tube
T1235-xxxG-TR	T1235xxxG	1.5 g	1000	Tape & reel

Note: xxx = voltage, yy = sensitivity, z = type

Fig. 1: Maximum power dissipation versus RMS on-state current (full cycle).

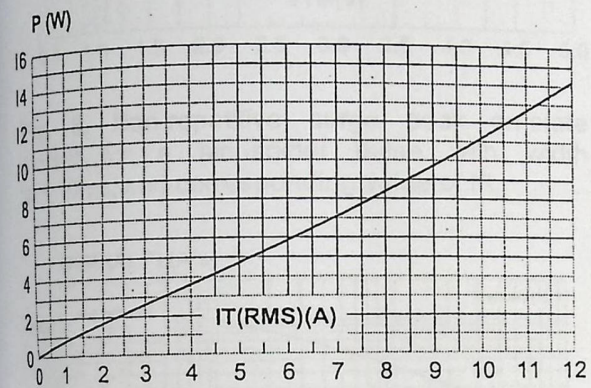


Fig. 2-2: RMS on-state current versus ambient temperature (printed circuit board FR4, copper thickness: 35 μ m), full cycle.

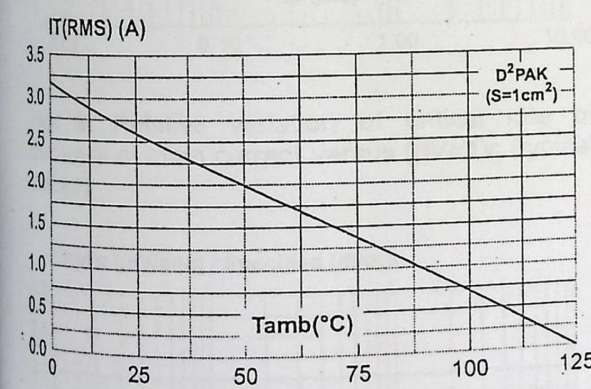


Fig. 2-1: RMS on-state current versus case temperature (full cycle).

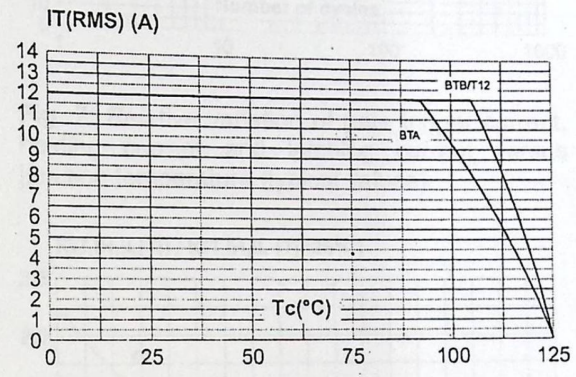


Fig. 3: Relative variation of thermal impedance versus pulse duration.

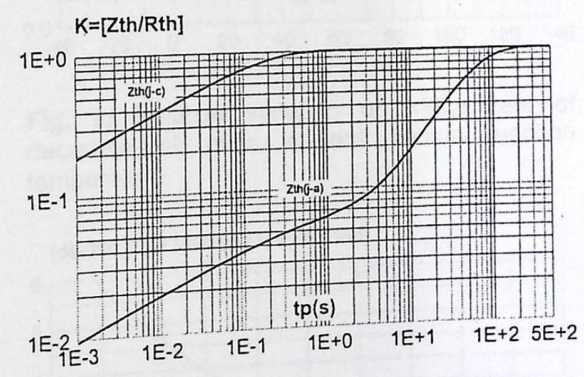


Fig. 4: On-state characteristics (maximum values).

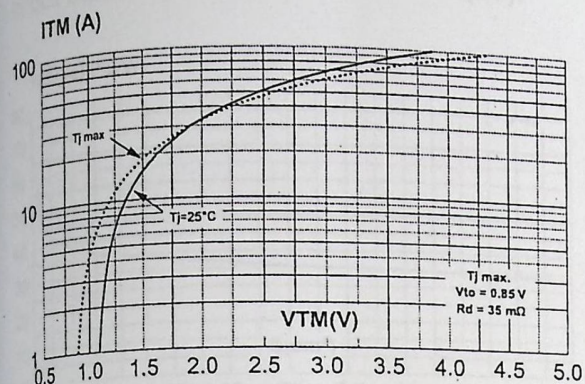


Fig. 6: Non-repetitive surge peak on-state current for a sinusoidal pulse with width $t_p < 10\text{ms}$, and corresponding value of I^2t .

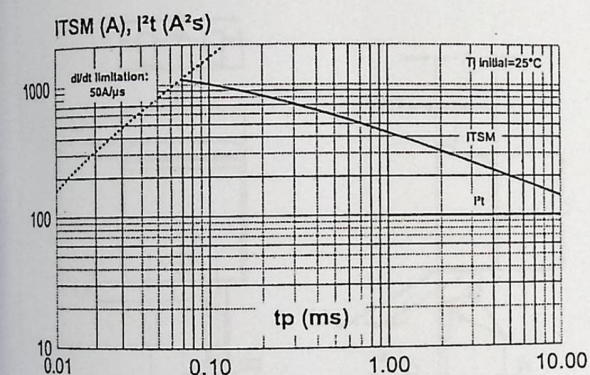


Fig. 8: Relative variation of critical rate of decrease of main current versus $(dV/dt)_c$ (typical values).

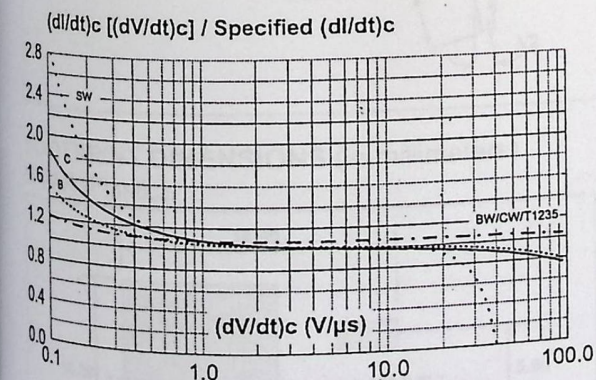


Fig. 5: Surge peak on-state current versus number of cycles.

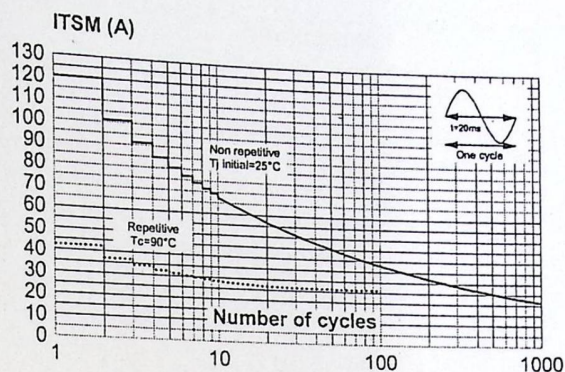


Fig. 7: Relative variation of gate trigger current, holding current and latching current versus junction temperature (typical values).

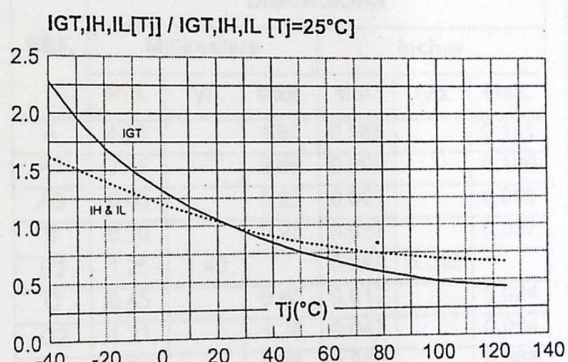


Fig. 9: Relative variation of critical rate of decrease of main current versus junction temperature.

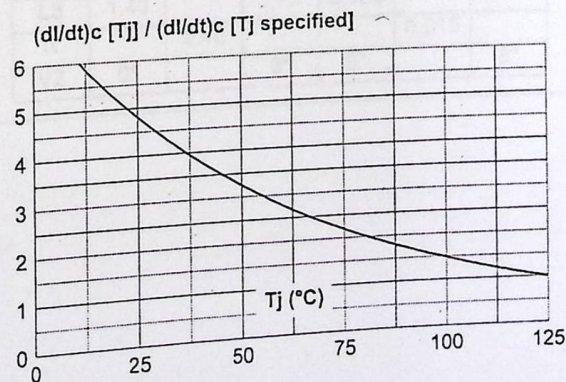
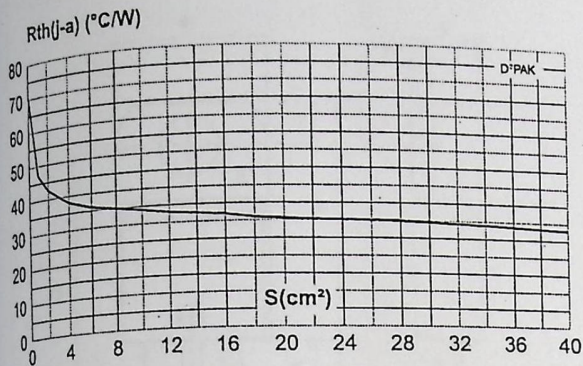
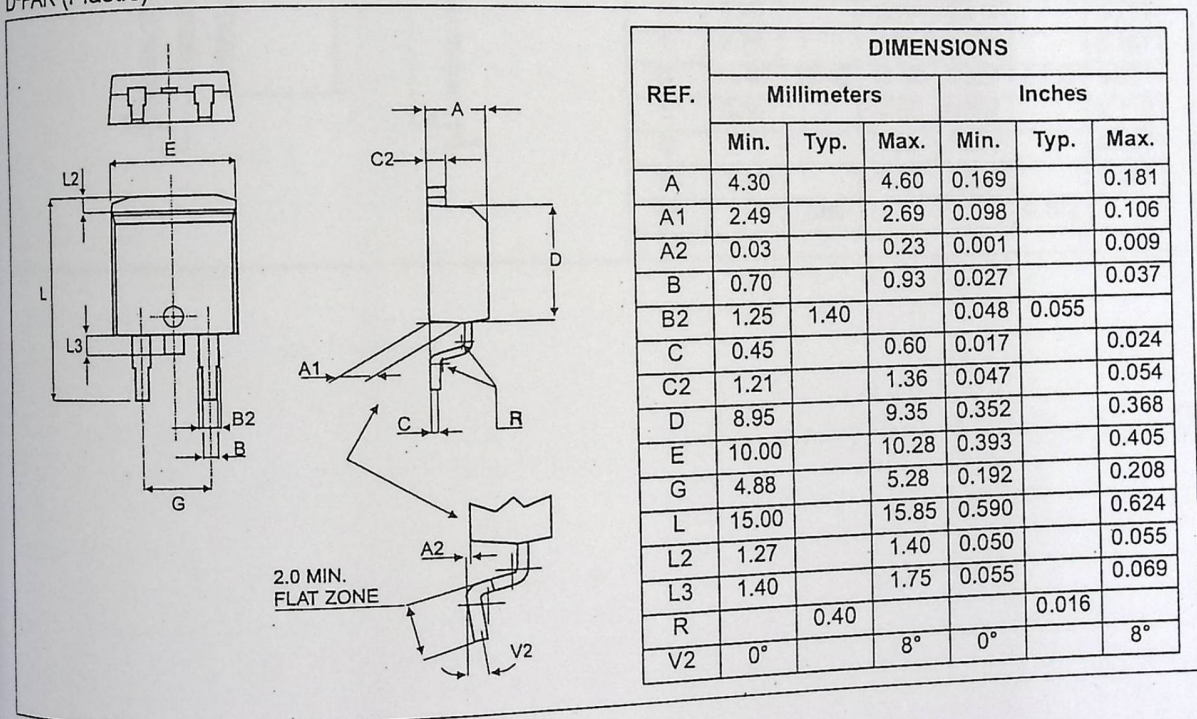


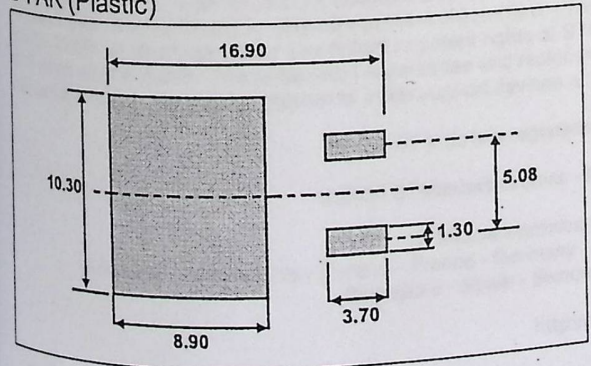
Fig. 10: D²PAK Thermal resistance junction to ambient versus copper surface under tab (printed circuit board FR4, copper thickness: 35 μm).



PACKAGE MECHANICAL DATA
D²PAK (Plastic)



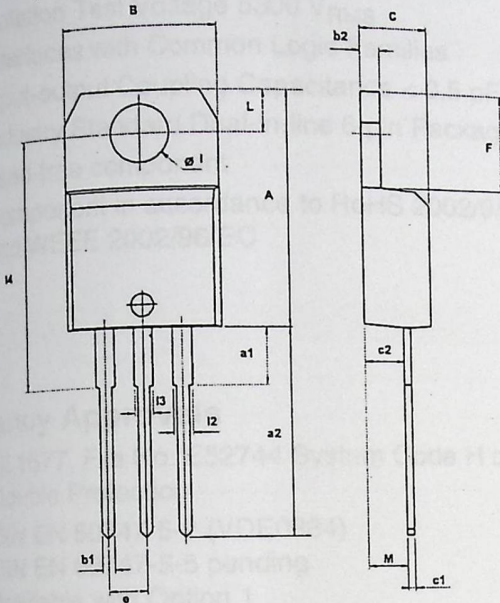
FOOTPRINT DIMENSIONS (in millimeters)
D²PAK (Plastic)



BTA/BTB12 and T12 Series

PACKAGE MECHANICAL DATA

TO-220AB / TO-220AB Ins.



REF.	DIMENSIONS					
	Millimeters			Inches		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A	15.20		15.90	0.598		0.625
a1		3.75			0.147	
a2	13.00		14.00	0.511		0.551
B	10.00		10.40	0.393		0.409
b1	0.61		0.88	0.024		0.034
b2	1.23		1.32	0.048		0.051
C	4.40		4.60	0.173		0.181
c1	0.49		0.70	0.019		0.027
c2	2.40		2.72	0.094		0.107
e	2.40		2.70	0.094		0.106
F	6.20		6.60	0.244		0.259
I	3.75		3.85	0.147		0.151
I4	15.80	16.40	16.80	0.622	0.646	0.661
L	2.65		2.95	0.104		0.116
I2	1.14		1.70	0.044		0.066
I3	1.14		1.70	0.044		0.066
M		2.60			0.102	

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

© The ST logo is a registered trademark of STMicroelectronics

© 2000 STMicroelectronics - Printed in Italy - All Rights Reserved

STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco
Singapore - Spain - Sweden - Switzerland - United Kingdom

<http://www.st.com>



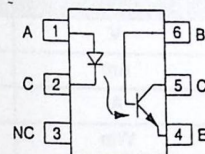
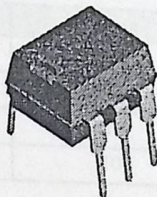
4N25/ 4N26/ 4N27/ 4N28

Vishay Semiconductors

Optocoupler, Phototransistor Output, With Base Connection

Features

- Isolation Test Voltage 5300 V_{RMS}
- Interfaces with Common Logic Families
- Input-output Coupling Capacitance < 0.5 pF.
- Industry Standard Dual-in-line 6-pin Package
- Lead-free component
- Component in accordance to RoHS 2002/95/EC and WEEE 2002/96/EC



1179004



Agency Approvals

- UL1577, File No. E52744 System Code H or J, Double Protection
- DIN EN 60747-5-2 (VDE0884)
DIN EN 60747-5-5 pending
Available with Option 1

Applications

AC Mains Detection
Reed relay driving
Switch Mode Power Supply Feedback
Telephone Ring Detection
Logic Ground Isolation
Logic Coupling with High Frequency Noise Rejection

Description

The 4N25 family is an Industry Standard Single Channel Phototransistor Coupler. This family includes the 4N25/ 4N26/ 4N27/ 4N28. Each optocoupler consists of gallium arsenide infrared LED and a silicon NPN phototransistor.

These couplers are Underwriters Laboratories (UL) listed to comply with a 5300 V_{RMS} isolation test voltage. This isolation performance is accomplished through special Vishay manufacturing process.

Compliance to DIN EN 60747-5-2(VDE0884)/ DIN EN 60747-5-5 pending partial discharge isolation specification is available by ordering option1.

These isolation processes and the Vishay ISO9001 quality program results in the highest isolation performance available for a commercial plastic phototransistor optocoupler.

The devices are also available in lead formed configuration suitable for surface mounting and are available either on tape and reel, or in standard tube shipping containers.

Note:

For additional design information see Application Note 45 Normalized Curves

Order Information

Part	Remarks
4N25	CTR > 20 %, DIP-6
4N26	CTR > 20 %, DIP-6
4N27	CTR > 10 %, DIP-6
4N28	CTR > 10 %, DIP-6
4N25-X006	CTR > 20 %, DIP-6 400 mil (option 6)
4N25-X007	CTR > 20 %, SMD-6 (option 7)
4N25-X009	CTR > 20 %, SMD-6 (option 9)
4N26-X006	CTR > 20 %, DIP-6 400 mil (option 6)
4N26-X007	CTR > 20 %, SMD-6 (option 7)
4N26-X009	CTR > 20 %, SMD-6 (option 9)
4N27-X007	CTR > 10 %, SMD-6 (option 7)
4N27-X009	CTR > 10 %, SMD-6 (option 9)
4N28-X009	CTR > 10 %, SMD-6 (option 9)

For additional information on the available options refer to Option Information.

www.vishay.com

1



Absolute Maximum Ratings

$T_{amb} = 25^{\circ}\text{C}$, unless otherwise specified

Excesses in excess of the absolute Maximum Ratings can cause permanent damage to the device. Functional operation of the device is not implied at these or any other conditions in excess of those given in the operational sections of this document. Exposure to absolute Maximum Rating for extended periods of the time can adversely affect reliability.

Input

Parameter	Test condition	Symbol	Value	Unit
Reverse voltage		V_R	6.0	V
Forward current		I_F	60	mA
Surge current	$t < 10 \mu\text{s}$	I_{FSM}	2.5	A
Power dissipation		P_{diss}	100	mW

Output

Parameter	Test condition	Symbol	Value	Unit
Collector-emitter breakdown voltage		V_{CEO}	70	V
Emitter-base breakdown voltage		V_{EBO}	7.0	V
Collector current		I_C	50	mA
Collector current	$t < 1.0 \text{ ms}$	I_C	100	mA
Power dissipation		P_{diss}	150	mW

Coupler

Parameter	Test condition	Symbol	Value	Unit
Isolation test voltage		V_{ISO}	5300	V_{RMS}
Creepage			≥ 7.0	mm
Clearance			≥ 7.0	mm
Isolation thickness between emitter and detector			≥ 0.4	mm
Comparative tracking index	DIN IEC 112/VDE0303, part 1		175	
Isolation resistance	$V_{IO} = 500 \text{ V}, T_{amb} = 25^{\circ}\text{C}$	R_{IO}	10^{12}	Ω
	$V_{IO} = 500 \text{ V}, T_{amb} = 100^{\circ}\text{C}$	R_{IO}	10^{11}	Ω
Storage temperature		T_{stg}	- 55 to + 150	$^{\circ}\text{C}$
Operating temperature		T_{amb}	- 55 to + 100	$^{\circ}\text{C}$
Junction temperature		T_j	100	$^{\circ}\text{C}$
Soldering temperature	max. 10 s, dip soldering: distance to seating plane $\geq 1.5 \text{ mm}$	T_{sld}	260	$^{\circ}\text{C}$



Absolute Maximum Ratings

at 25 °C, unless otherwise specified
 Excesses in excess of the absolute Maximum Ratings can cause permanent damage to the device. Functional operation of the device is not guaranteed at these or any other conditions in excess of those given in the operational sections of this document. Exposure to absolute Maximum Rating for extended periods of the time can adversely affect reliability.

Input

Parameter	Test condition	Symbol	Value	Unit
Reverse voltage		V_R	6.0	V
Forward current		I_F	60	mA
Pulse current	$t < 10 \mu s$	I_{FSM}	2.5	A
Power dissipation		P_{diss}	100	mW

Output

Parameter	Test condition	Symbol	Value	Unit
Collector-emitter breakdown voltage		V_{CEO}	70	V
Emitter-base breakdown voltage		V_{EBO}	7.0	V
Collector current		I_C	50	mA
Collector current	$t < 1.0 ms$	I_C	100	mA
Power dissipation		P_{diss}	150	mW

Coupler

Parameter	Test condition	Symbol	Value	Unit
Isolation test voltage		V_{iso}	2500	V
Creepage			1.27	mm
Clearance			1.27	mm
Isolation thickness between emitter and detector			1.27	mm
Comparative tracking index	DIN EN 60112, Part 1		100%	
Isolation resistance	$V_{iso} = 500 V, T_{amb} = 25^\circ C$ $V_{iso} = 500 V, T_{amb} = 125^\circ C$	R_{iso} R_{iso}	$10^{10} \Omega$ $10^9 \Omega$	
Storage temperature		T_{stg}	-55 to 150	°C
Operating temperature		T_{amb}	-55 to 125	°C
Junction temperature		T_J	125	°C
Guiding temperature	max. 10 s, 100 mm distance to heat sink $\geq 1.5 mm$	T_{con}	125	°C

Parameter	Unit
	%
	%
	%
	%



4N25/ 4N26/ 4N27/ 4N28

Vishay Semiconductors

Electrical Characteristics

$T_{amb} = 25^{\circ}\text{C}$, unless otherwise specified

Minimum and maximum values are testing requirements. Typical values are characteristics of the device and are the result of engineering evaluation. Typical values are for information only and are not part of the testing requirements.

Input

Parameter	Test condition	Symbol	Min	Typ.	Max	Unit
Forward voltage ¹⁾	$I_F = 50\text{ mA}$	V_F		1.3	1.5	V
Reverse current ¹⁾	$V_R = 3.0\text{ V}$	I_R		0.1	100	μA
Capacitance	$V_R = 0\text{ V}$	C_O		25		pF

¹⁾ Indicates JEDEC registered values

Output

Parameter	Test condition	Part	Symbol	Min	Typ.	Max	Unit
Collector-base breakdown voltage ¹⁾	$I_C = 100\text{ }\mu\text{A}$		BV_{CBO}	70			V
Collector-emitter breakdown voltage ¹⁾	$I_C = 1.0\text{ mA}$		BV_{CEO}	30			V
Emitter-collector breakdown voltage ¹⁾	$I_E = 100\text{ }\mu\text{A}$		BV_{ECO}	7.0			V
$I_{CEO}(\text{dark})$ ¹⁾	$V_{CE} = 10\text{ V}$, (base open)	4N25			5.0	50	nA
		4N26			5.0	50	nA
		4N27			5.0	50	nA
		4N28			10	100	nA
$I_{CBO}(\text{dark})$ ¹⁾	$V_{CB} = 10\text{ V}$, (emitter open)				2.0	20	nA
Collector-emitter capacitance	$V_{CE} = 0$		C_{CE}		6.0		pF

¹⁾ Indicates JEDEC registered values

Coupler

Parameter	Test condition	Part	Symbol	Min	Typ.	Max	Unit
Isolation voltage ¹⁾	Peak, 60 Hz	4N25	V_{IO}	2500			V
		4N26	V_{IO}	1500			V
		4N27	V_{IO}	1500			V
		4N28	V_{IO}	500			V
			$V_{CE(sat)}$			0.5	V
Saturation voltage, collector-emitter	$I_{CE} = 2.0\text{ mA}$, $I_F = 50\text{ mA}$		R_{IO}	100			$\text{G}\Omega$
Resistance, input output ¹⁾	$V_{IO} = 500\text{ V}$		C_{IO}		0.5		pF
Capacitance (input-output)	$f = 1.0\text{ MHz}$						

¹⁾ Indicates JEDEC registered values

Current Transfer Ratio

Parameter	Test condition	Part	Symbol	Min	Typ.	Max	Unit
DC Current Transfer Ratio ¹⁾	$V_{CE} = 10\text{ V}$, $I_F = 10\text{ mA}$	4N25	CTR_{DC}	20	50		%
		4N26	CTR_{DC}	20	50		%
		4N27	CTR_{DC}	10	30		%
		4N28	CTR_{DC}	10	30		%

¹⁾ Indicates JEDEC registered value

www.vishay.com



Switching Characteristics

Parameter	Test condition	Symbol	Min	Typ.	Max	Unit
Rise and fall times	$V_{CE} = 10\text{ V}$, $I_F = 10\text{ mA}$, $R_L = 100\ \Omega$	t_r , t_f		2.0		μs

Typical Characteristics ($T_{amb} = 25\ ^\circ\text{C}$ unless otherwise specified)

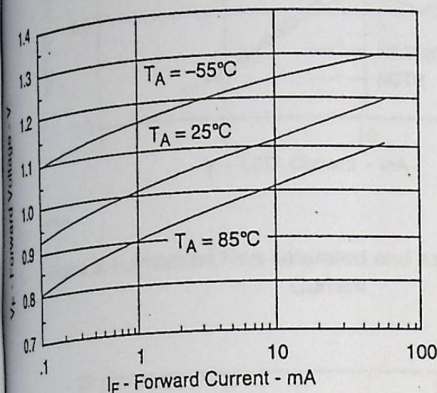
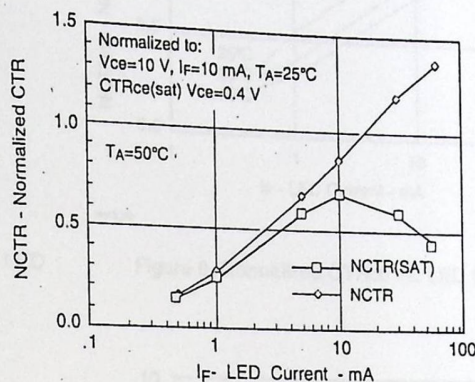


Figure 1. Forward Voltage vs. Forward Current



4N25_03

Figure 3. Normalized Non-saturated and Saturated CTR vs. LED Current

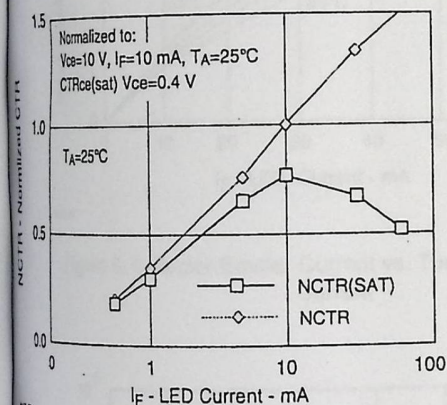
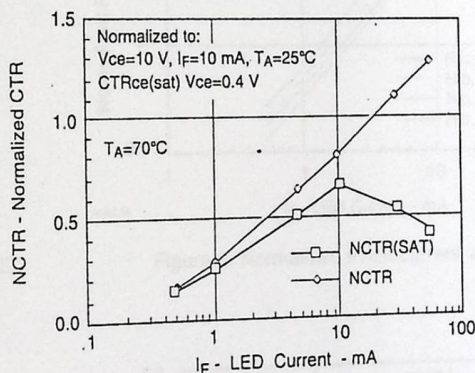


Figure 2. Normalized Non-Saturated and Saturated CTR vs. LED Current



4N25_04

Figure 4. Normalized Non-saturated and saturated CTR vs. LED Current

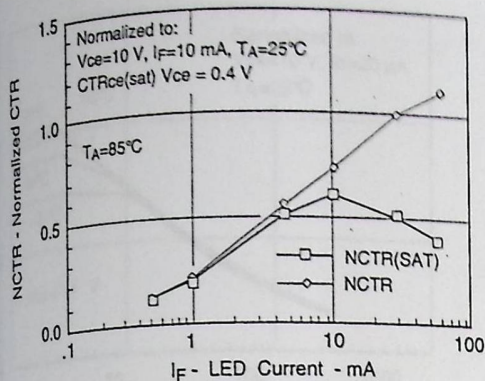


Figure 5. Normalized Non-saturated and saturated CTR vs. LED Current

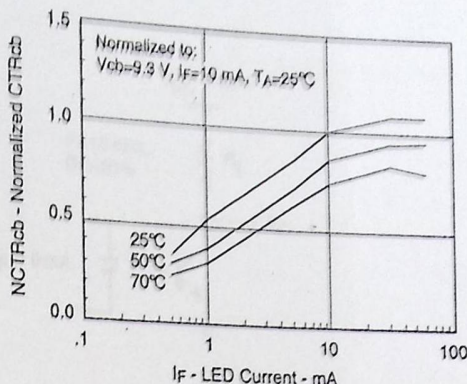


Figure 8. Normalized CTRcb vs. LED Current and Temp.

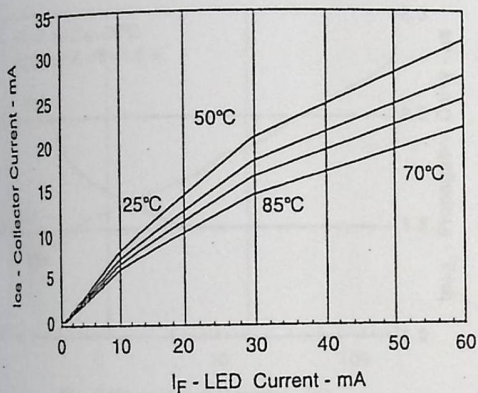


Figure 6. Collector-Emitter Current vs. Temperature and LED Current

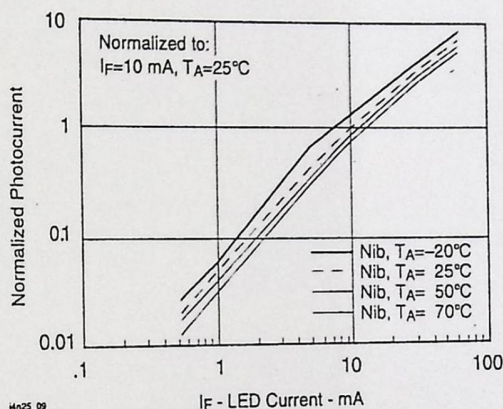


Figure 9. Normalized Photocurrent vs. I_F and Temp.

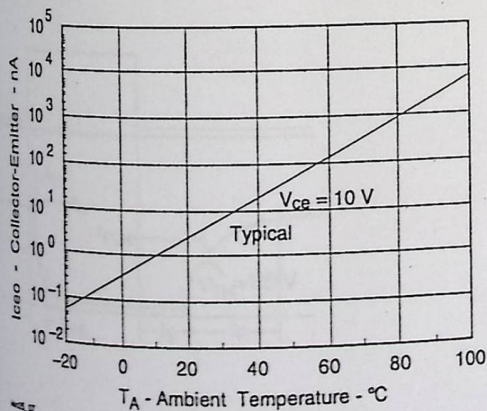


Figure 7. Collector-Emitter Leakage Current vs. Temp.

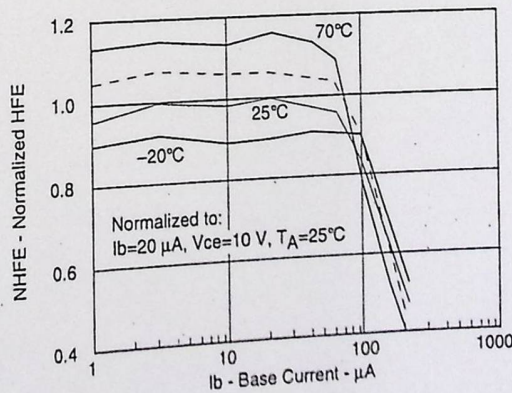


Figure 10. Normalized Non-saturated HFE vs. Base Current and Temperature

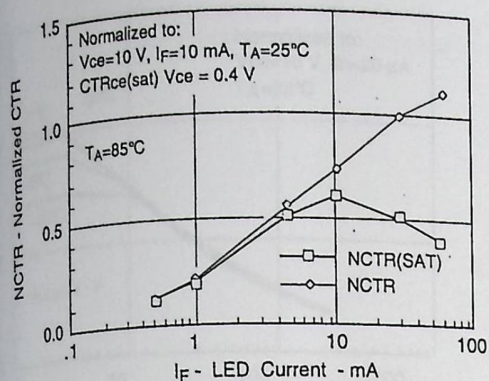


Figure 5. Normalized Non-saturated and saturated CTR vs. LED Current

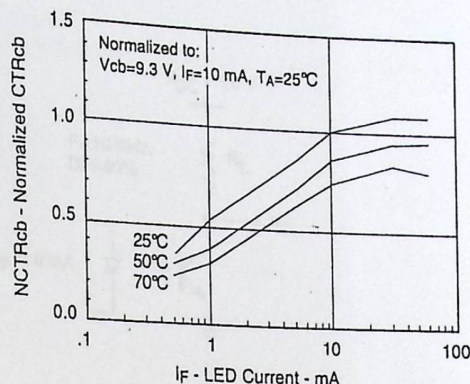


Figure 8. Normalized CTRcb vs. LED Current and Temp.

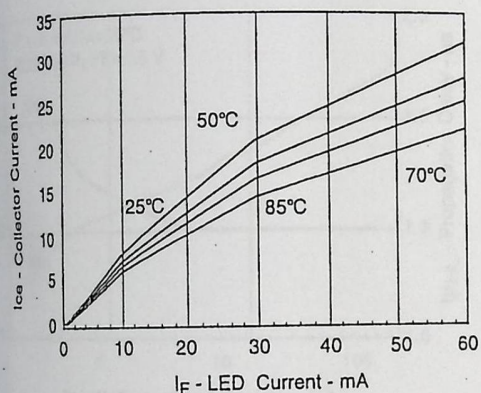


Figure 6. Collector-Emitter Current vs. Temperature and LED Current

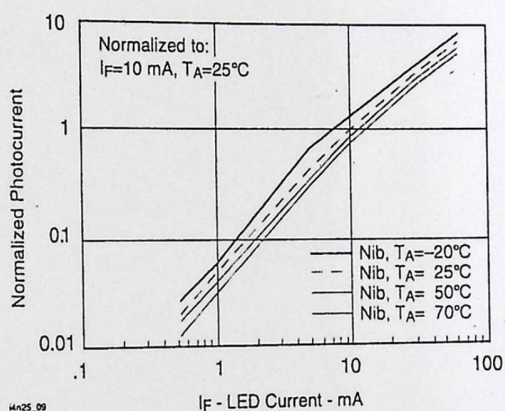


Figure 9. Normalized Photocurrent vs. I_F and Temp.

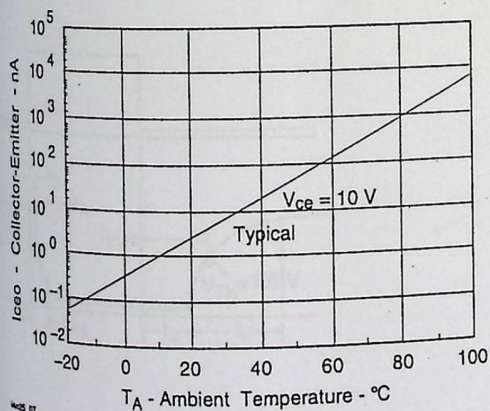


Figure 7. Collector-Emitter Leakage Current vs. Temp.

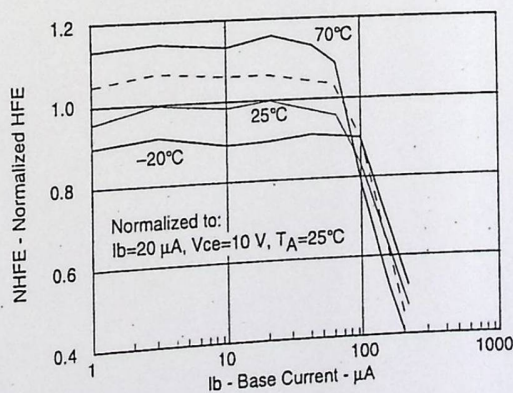


Figure 10. Normalized Non-saturated HFE vs. Base Current and Temperature

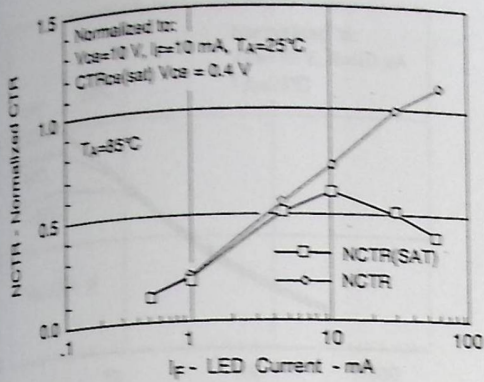


Figure 5. Normalized Non-saturated and saturated CTR vs. LED Current

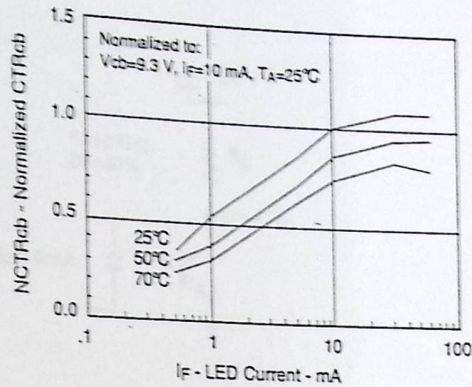


Figure 8. Normalized CTRcb vs. LED Current and Temp.

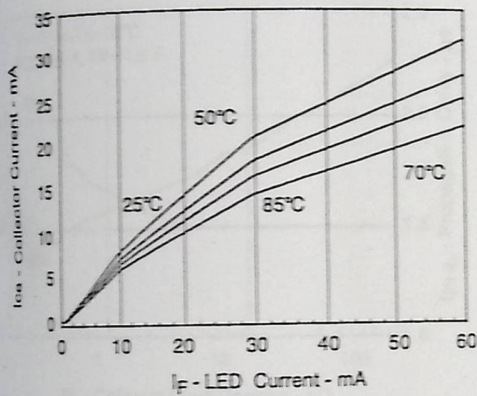


Figure 6. Collector-Emitter Current vs. Temperature and LED Current

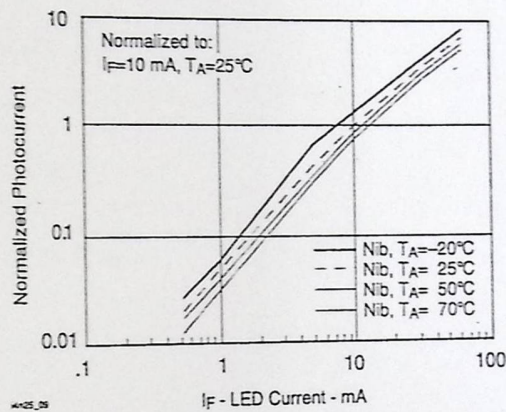


Figure 9. Normalized Photocurrent vs. I_F and Temp.

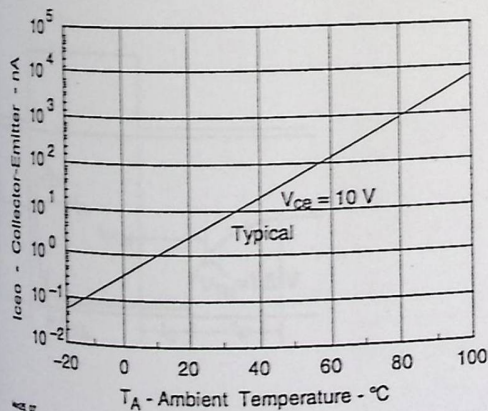


Figure 7. Collector-Emitter Leakage Current vs. Temp.

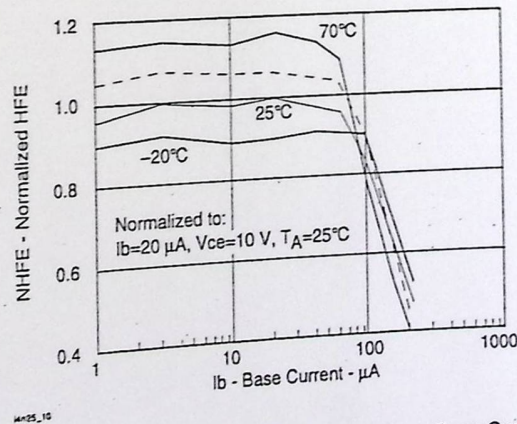


Figure 10. Normalized Non-saturated HFE vs. Base Current and Temperature

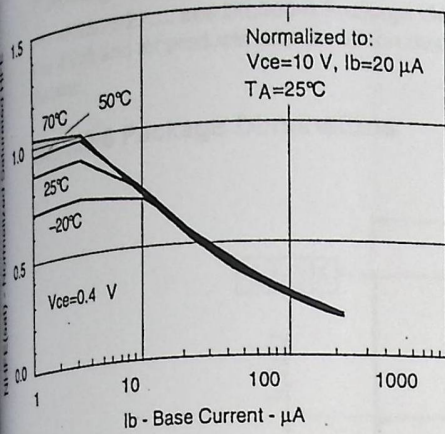


Figure 11. Normalized HFE vs. Base Current and Temp.

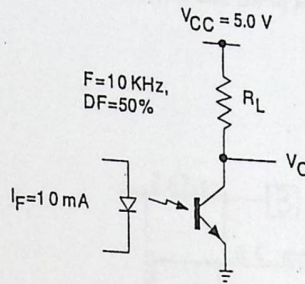


Figure 14. Switching Schematic

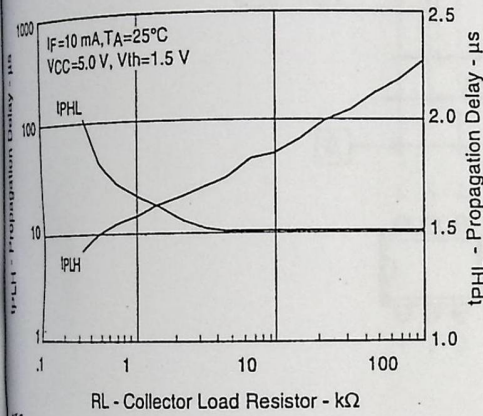


Figure 12. Propagation Delay vs. Collector Load Resistor

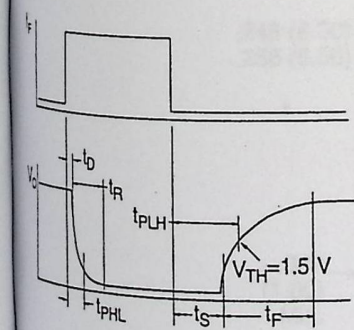


Figure 13. Switching Timing



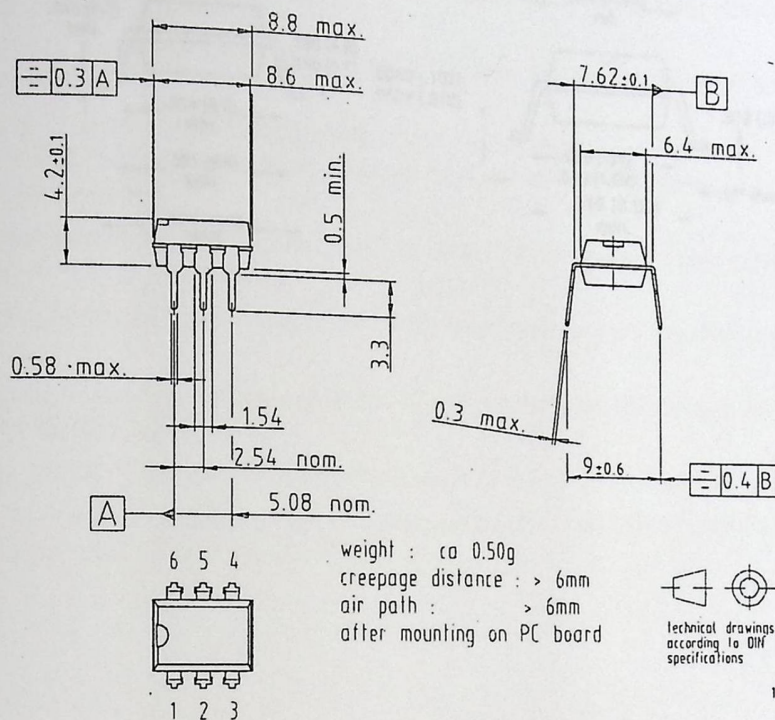
4N25/ 4N26/ 4N27/ 4N28

Vishay Semiconductors

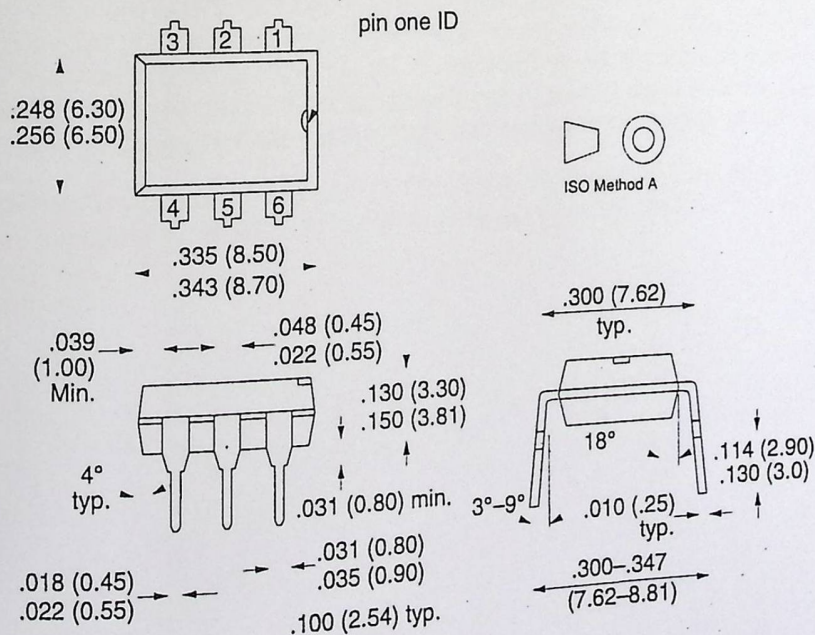
Package Dimensions in Inches (mm)

For 4N25/26/27..... see DIL300-6 Package dimension in the Package Section.
For 4N28 and for products with an option designator (e.g. 4N25-X001 or 4N26-X007)..... see DIP-6 Package dimensions in the Package Section.

DIL300-6 Package Dimensions



DIP-6 Package Dimensions



1178004

www.vishay.com



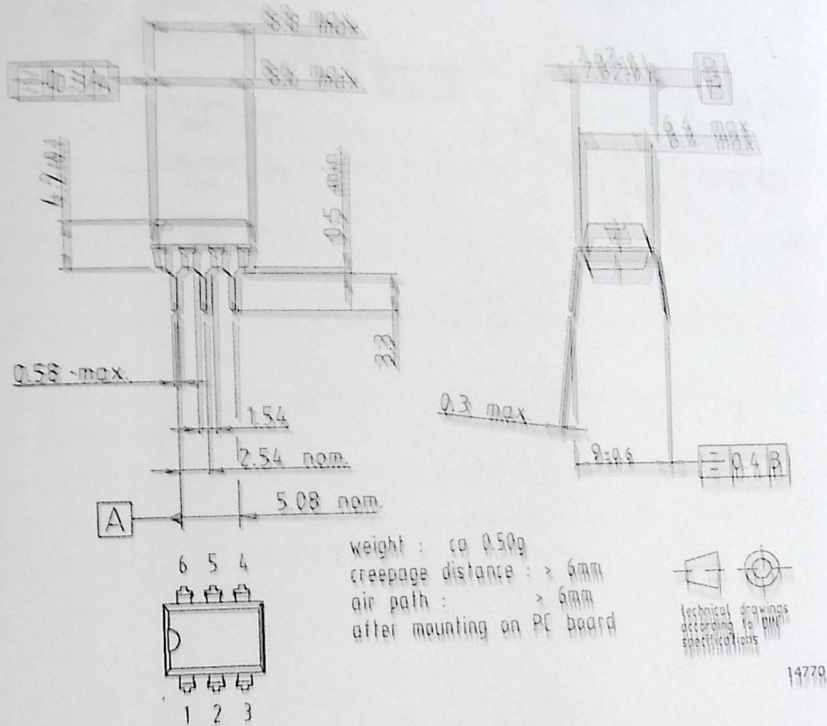
4N25/ 4N26/ 4N27/ 4N28

Vishay Semiconductors

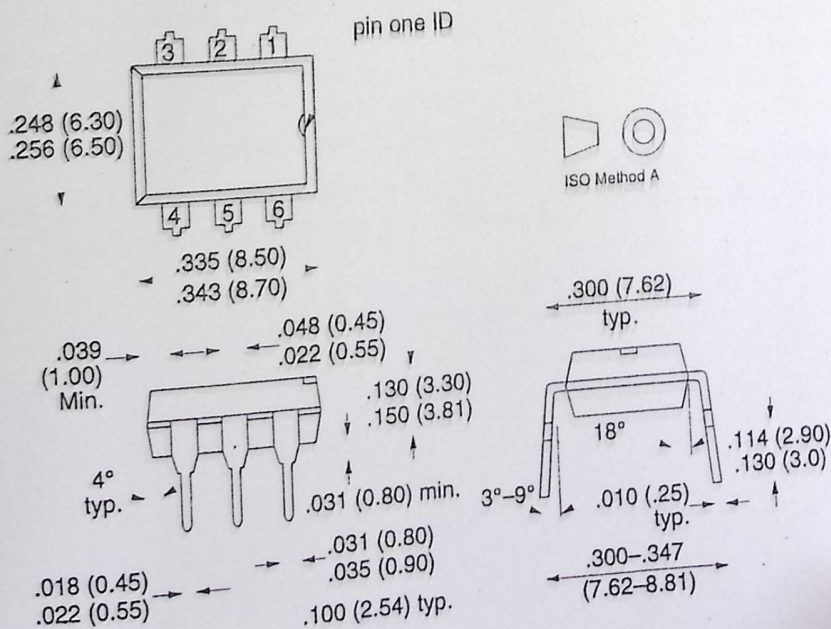
Package Dimensions in Inches (mm)

For 4N25/26/27..... see DIL300-6 Package dimension in the Package Section.
For 4N28 and for products with an option designator (e.g. 4N25-X001 or 4N26-X007)..... see DIP-6 Package dimensions in the Package Section.

DIL300-6 Package Dimensions



DIP-6 Package Dimensions



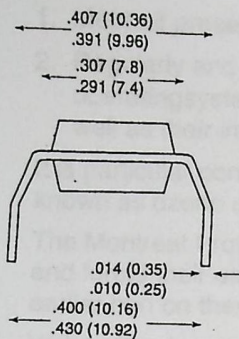
1178004

4N25/ 4N26/ 4N27/ 4N28

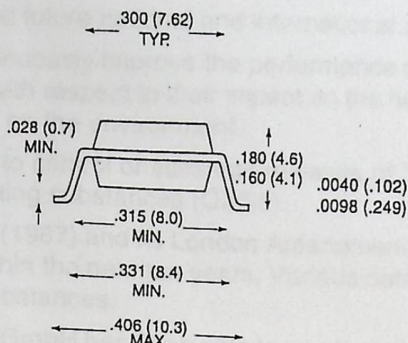
Vishay Semiconductors



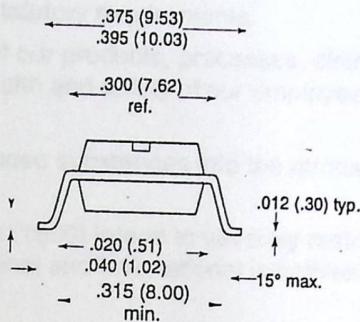
Option 6



Option 7



Option 9



18450

1. Annex A, B and C of transitional substances of the Montreal Protocol and the London Amendment respectively
 2. Class I and II ozone depleting substances of the Clean Air Act Amendments of 1990 by the Environmental Protection Agency (EPA) in the USA
 3. Council Decision 89/540/EEC and 91/691/EEC Annex A, B and C substances respectively (Annex B only)
- Vishay Semiconductor GmbH can certify that our semiconductor products are not manufactured with any of the foregoing substances and do not contain such substances.

We reserve the right to make changes to our products without notice and may do so without further notice.

Parameters can vary in different applications. All drawing parameters must be verified for each customer application by the customer. Should the buyer use Vishay Semiconductors' products for any unapproved or unauthorized application, the buyer shall indemnify Vishay Semiconductors against all claims, costs, damages, and expenses, arising out of, directly or indirectly, any claim in respect of damage, injury or death associated with such unauthorized or unapproved use.

Vishay Semiconductor GmbH, P.O.B. 3635, D-74025 Heidenheim, Germany
Telephone: 49 (0)7151 67 2331, Fax number: 49 (0)7151 67 2495



Ozone Depleting Substances Policy Statement

It is the policy of Vishay Semiconductor GmbH to

1. Meet all present and future national and international statutory requirements.
2. Regularly and continuously improve the performance of our products, processes, distribution and operating systems with respect to their impact on the health and safety of our employees and the public, as well as their impact on the environment.

It is particular concern to control or eliminate releases of those substances into the atmosphere which are known as ozone depleting substances (ODSs).

The Montreal Protocol (1987) and its London Amendments (1990) intend to severely restrict the use of ODSs and forbid their use within the next ten years. Various national and international initiatives are pressing for an earlier ban on these substances.

Vishay Semiconductor GmbH has been able to use its policy of continuous improvements to eliminate the use of ODSs listed in the following documents.

1. Annex A, B and list of transitional substances of the Montreal Protocol and the London Amendments respectively
2. Class I and II ozone depleting substances in the Clean Air Act Amendments of 1990 by the Environmental Protection Agency (EPA) in the USA
3. Council Decision 88/540/EEC and 91/690/EEC Annex A, B and C (transitional substances) respectively.

Vishay Semiconductor GmbH can certify that our semiconductors are not manufactured with ozone depleting substances and do not contain such substances.

We reserve the right to make changes to improve technical design
and may do so without further notice.

Parameters can vary in different applications. All operating parameters must be validated for each customer application by the customer. Should the buyer use Vishay Semiconductors products for any unintended or unauthorized application, the buyer shall indemnify Vishay Semiconductors against all claims, costs, damages, and expenses, arising out of, directly or indirectly, any claim of personal damage, injury or death associated with such unintended or unauthorized use.

Vishay Semiconductor GmbH, P.O.B. 3535, D-74025 Heilbronn, Germany
Telephone: 49 (0)7131 67 2831, Fax number: 49 (0)7131 67 2423

References

Books

- Andy Mcnab, Remote Control, Ballantine Publishing Group , 1997.
- Ivan G Stearne, How to design\build remote control device, Prentice-Hall,1998.
- John Wiley Son Inc, Infrared Technology, Published Simultaneously Canada, 2005.

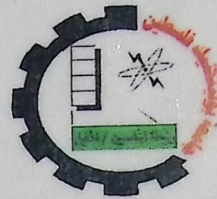
Papers:

- Asm's Abu Salah and Mahmoud Ewawee , "PC remote control system", Palestine polytechnic university ,Hebron ,Palestine ,2001
- Hani Mis'ef and Mahmoud Shratch, "Infrared general purposes Socket" ,Birzit University ,Ramallah ,Palestine ,2001

Internet:

- <http://www.epanorama.net/links /irremote.html>
- <http://winlirc.sourceforge.net>
- <http://www.lirc.org/recievers.html>
- <http://www.ee.washington.edu/ciruit archive/circuits/PCIR/welcome.html>
- <http://lnx.manoweb.com/lirc/?partType=section&partName=winlirc>
- <http://www.4p8.com/eric.brasseur/receiv.html>

Palestine Polytechnic University



College of Engineering & Technology
Electrical & Computer Engineering

Graduation Project

**A Personal Computer Based General Purpose Control System
Using General Remote Control**

Project Team

Ali M.Salameh
Omar G.Al-Rjoup
Hussein R.Al-Manasra

Project Supervisor
Eng.Mazan K.Zalloum

Hebron – Palestine
May, 2006