SPY ROBOT

A Robot for Wireless Video and Audio Transmission

A Project Report

Submitted By

K.PRABAGARANE	**-**	**71204105036**
J.SATHEESH KUMAR	**-**	**71204105048**
N.SOLAIRAJ	**-**	**71204105052**

in partial fulfillment for the award of the degree

of

Bachelor of Engineering

In

Electrical & Electronics Engineering

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006

APRIL – 2008

ANNA UNIVERSITY :: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report entitled **"SPY ROBOT - A ROBOT FOR WIRELESS VIDEO AND AUDIO TRANSMISSION"** is a bonafide work of

K.PRABAGARANE	-	71204105036
J.SATHEESH KUMAR	-	71204105048
N.SOLAIRAJ	-	71204105052

Who carried out the project under my supervision.

Signature of the DEAN/HOD
(Prof.K.Ragupathy Subramanian)
DEAN/HOD

Signature of the supervisor
(Mr. C. Udhaya Shankar, M.Tech)
Lecturer

Certified that the candidate with university Register No. 71204105036 , 71204105048 , 71204105052 was examined in project viva voce Examination held on 17.04.08

Internal Examiner

17/4/08
External Examiner

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006

ABSTRACT

This project "SPY ROBOT" deals with the computer controlled robot for the transmission of video and audio signals. The robot placed in a hazardous environment is controlled by a human being in a secure place.

The project consists of two sections namely control section and robot section. The control section transmits RF signals to control the robot. The robot section consists of a RF receiver which receives the control signals . The control signals are fed into the pre programmable microcontroller which drives the driver circuit and hence the movement of robot is controlled.

The video camera module placed in the robot captures the video and audio signals in the hazardous environment and transmits it to the control section. The received video and audio signals are fed into the TV tuner card and viewed in the computer with the help of visual basic front end.

The metallic detector placed in the robot senses the presence of the bomb in its path. If there is any such detection the buzzer gives alarm and also the robot is stopped.

ACKNOWLEDGEMENT

The completion of our project can be attributed to the combined efforts made by us and the contribution made in one form or the other by the individuals we hereby acknowledge.

We are highly privileged to thank **Dr.Joseph V.Thanikal**, Principal, Kumaraguru College of Technology for providing facilities to do this project.

We express our heart felt gratitude and thanks to the Dean/ HOD of Electrical & Electronics Engineering, **Prof.K.Regupathy Subramanian, B.E(Hons), M.Sc.** for encouraging us and for being with us right from beginning of the project and guiding us at every step.

We wish to place on record our deep sense of gratitude and profound thanks to our guide **Mr.C.Udhayashankar, M.Tech** Electrical and Electronics Engineering Department, for his valuable guidance, constant encouragement, continuous support and co-operation rendered throughout the project.

We are also thankful to our teaching and non-teaching staffs of Electrical and Electronics Engineering Department, for their kind help and encouragement.

Last but not least, we extend our sincere thanks to all our **parents and friends** who have contributed their ideas and encouraged us for completing the project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 NEED FOR THE PROJECT:

In the present world the field of Robotics is improving in leaps and bounds. There are enormous numbers of application where the Human beings are replaced by Robots. There's also a need to free up human from hazardous and unsafe sites and utilize well designed telerobot for those purpose. This is the vital driving fact behind our project.

1.2.DESIGN FACTORS :

- Proper lighting of the work space
- Camera with required resolution to extract dimensions
- The field of view should be large enough to accommodate the part
- Robot should have sufficient degrees of freedom to manipulate the camera.

1.3 OBJECTIVES AND FEATURES OF THE PROJECT:

In the project "Spy robot" has been developed for use in unsafe environment. This robot can move around in all possible direction and can transmit video from the survey field where it is positioned to the PC in far-off place.

- Video captured by the robot will be transmitted in air using video transmitter.
- The video signals will be captured by the antenna and fed into the pc via tuner card .
- There will be a VB screen which eliminates the need for having separate screen to view the transmitted video .
- The operator could see the survey field in the screen and guide the robot accordingly .

1.4 APPLICATIONS OF THE PROJECT:

Due to various problems encountered by human beings in dangerous areas there's desperately a need to employ robots in such sites. So Spy robot finds a real application in such place where human entry is unsafe. This Robot can be used in nuclear and chemical plants, near oil wells, as watch dog for security purpose and for military operations. It is also used to investigate the unexplored areas which may contain toxic gases and also used as space vehicle to capture image from other planets.

1.5 BLOCK DIAGRAM OF THE PROJECT

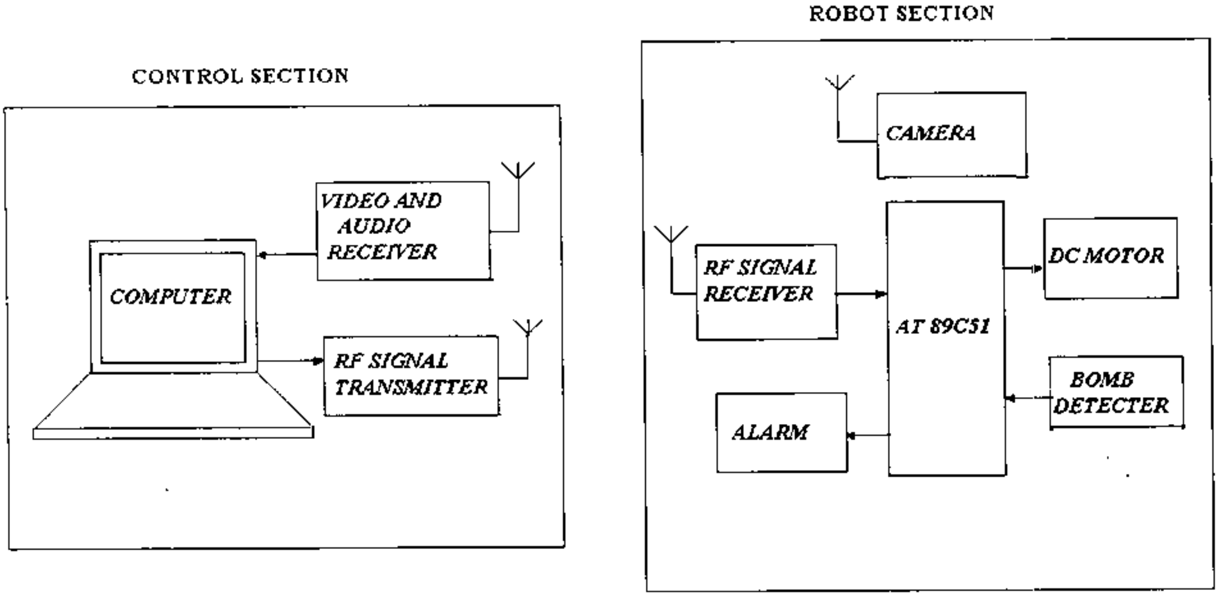


Fig 1.1 Block diagram

1.6 PROJECT OVERVIEW

Spyrobot is a telerobot where the operator in safe environment guides the robot in hazardous site.The overall view of the project is given below. The project consists of two main sections namely,

- Control Section &
- Robot Section.

CONTROL SECTION :

As the name suggests the entire operaton of the Robot is controlled by the operator using this section. Here Visual Basic (VB) is used as front end . The main components of this section are

- Front end ,
- Signal Transmitter ,
- Video and Audio Receiver.

FRONT END:

Visual Basic is used as front end in our project. The VB screen has buttons to control the movement of Robot in the survey field. It also has a screen to view the video captured by the Robot in its path. VB uses parallel port to send the signals to the transmitter ciruit to control the Robot.

The main advantage of using VB as font end are

User interactive.

- Coding easier.
- Easy to Debug and test output.

SIGNAL TRANSMITTER SECTION:

The Transmitter section transmits the signals to the Robot section. The input to this section is given through Parallel Port. The data is sent to robot section via RF wave at a frquency of 433 MHz.

The Transmitter section has two main components namely,

- Encoder (HT 640),
- RF Transmitter (TX 434).

Encoder :

The encoder used here is HT 640. It receives input from the parallel port of the PC. It converts the parallel input data to serial output and gives it to transmitter.

Transmitter:

The Transmitter used here is TWS 434. It transmits the serial data given to it from encoder, to robot secion in air via RF wave at 433 MHz.

VIDEO AND AUDIO RECEIVER:

This section receives the video of the survey field, sent by video transmitter circuit in Robot section. An antenna is used for this purpose. Its output is given to the PC via TV Tuner Card. From there it is fetched and displayed in the Front end of the VB screen.

ROBOT SECTION :

This is the core section of our project. This section receives the signals from the control section and acts accordingly. The main components of this section are given below

- Signal reception section,
- Microcontroller (89C51),
- Driver circuit,
- Bomb detection circuit,
- Camera

SIGNAL RECEPTION SECTION

The signal receiver section receives the signals transmitted from control section via RF wave. The data received is used for controlling the robot operation.

The Transmitter section has two main components namely,

- ✓ RF Receiver (RX 434),
- ✓ Decoder (HT 648L).

RF Receiver:

The Receiver used to receive RF signal is RWS 434. It receives the serial data via RF transmitted by the control section and gives it to the decoder module.

Decoder :

The Decoder unit used is HT 648. Input to decoder is given from the RF receiver module. It converts the serial input to parallel output and gives it to the microcontroller unit.

MICROCONTROLLER :

The heart of the Robot section is the microcontroller - ATMEL 89C51. Microcontroller controls the entire operation of robot and also the bomb detector.

Advantages of using microcontroller :

- ✓ Inbuilt storage,
- ✓ PCB size reduced,
- ✓ Low design cost

This module receives the input from the receiver section and processes the data.It controls the driver circuit and bomb detector. It has many features like On - Chip Flash Program Memory, On - Chip Data RAM, Watch dog timer etc. This is dealt in the later sections.

DRIVER CIRCUIT:

The driver circuit is used to control the inputs to two DC motors in the Robot thereby controlling the movement of the Robot. The driver circuit is controlled by the microcontroller. It performs five operations based on the input it receives. The operations are given below

- Forward
- Reverse
- Left
- Right
- Stop

BOMB DETECTION CIRCUIT :

There is a metallic detector in the Robot section. It senses the any bomb occurance in its path. If there is any bomb, the detector senses it, halts the robot and gives an continuous alarm (Buzzer). The sensor circuit also triggers the relay. The required operation such as bomb extraction can be implemented as per needs in the relay circuit in future.

VIDEO AND AUDIO TRANSMITTER SECTION :

The video transmission circuit is used to transmit the video captured by the camera, in the survey field to the control section to be seen in the front end. The captured video is given to the video transmitter circuit. From there it is transmited as Video signal in air. This signal is received in control section by using the receiver ie the antenna and viewed in VB screen.

2. PC INTERFACE AND CONTROL UNIT FOR ROBOT

2.1 SYSTEM DESCRIPTION OF CONTROL SECTION

2.1.1 HARDWARE UNIT:

The control section consists of three units,

- Visual basic (interface with parallel port),
- Encoder HT 640,
- Transmitter TX 434.

The control signals from the PC is given to encoder (HT640) through parallel port (LPT connector).The encoder encodes 8-Bit parallel data to serial data. The output from the encoder is given to the data input pin of transmitter TWS-434. The transmitter manipulates the 8-Bit digital code to the RF wave and it transmits the signals at a frequency of 434 MHz.

2.2 POWER SUPPLY UNIT

2.2.1 INTRODUCTION:

The present chapter introduces the operation of power supply circuits built using filters, rectifiers, and then voltage regulators. Starting with an ac voltage, a steady dc voltage is obtained by rectifying the ac voltage, then filtering to a dc level, and finally, regulating to obtain a desired fixed dc voltage. The regulation is usually obtained from an IC voltage regulator unit, which takes a dc voltage and provides a somewhat lower dc voltage, which remains the same even if the input dc voltage varies, or the output load connected to the dc voltage changes.

A block diagram containing the parts of a typical power supply and the voltage at various points in the unit is shown in fig . The ac voltage, typically 120 V rms, is connected to a transformer, which steps that ac voltage down to the level for the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation. A regulator circuit can use this dc input to provide a dc voltage that not only has much less ripple voltage but also remains the same dc value even if the input dc voltage varies somewhat, or the load connected to the output dc voltage changes. This voltage regulation is usually obtained using one of a number of popular voltage regulator IC units.

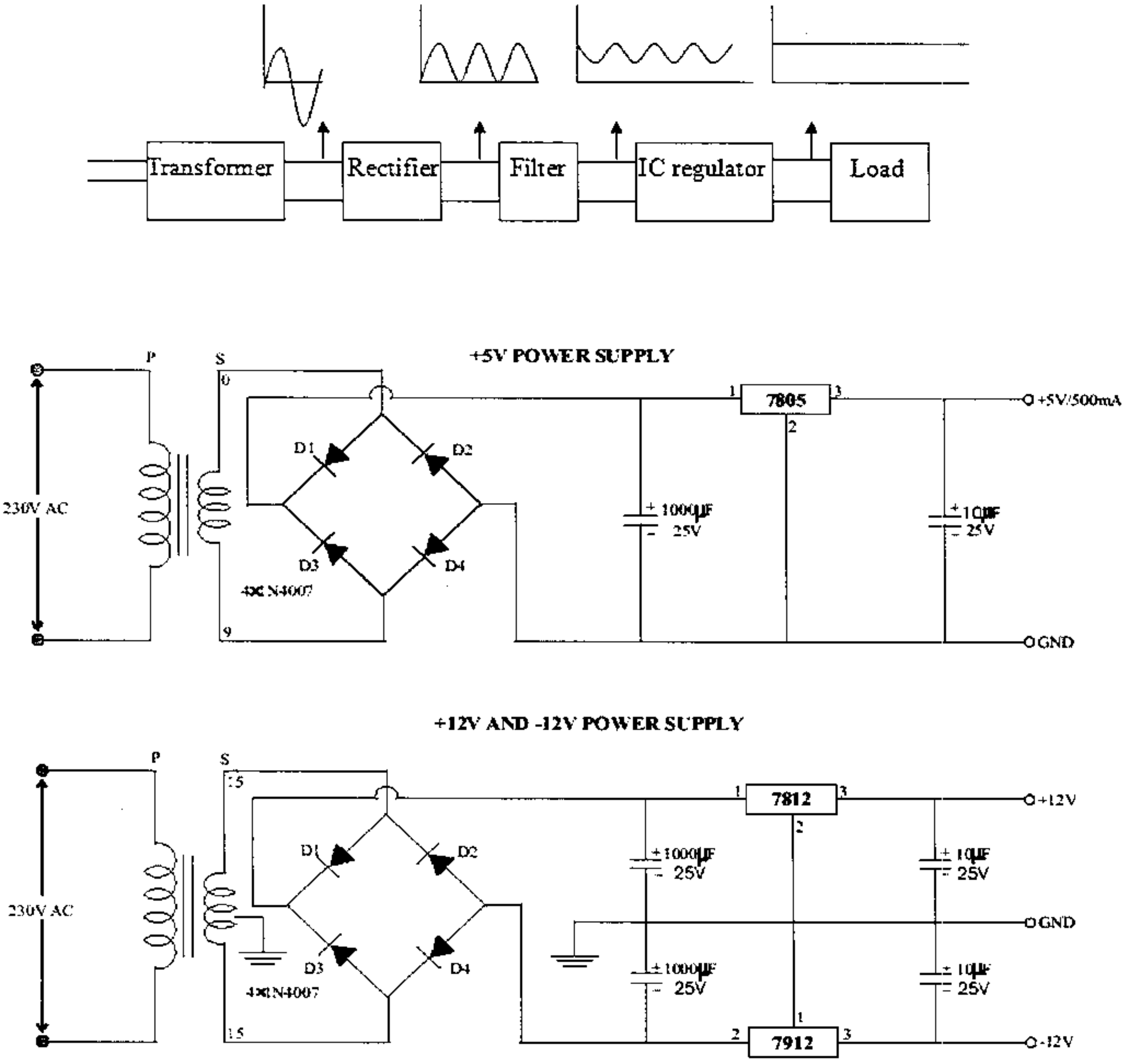


Fig 2.1 Power supply

2.3 VISUAL BASIC

2.3.1 Introduction:

Visual Basic is a programming environment from Microsoft in which a programmer uses a graphical user interface to choose and modify preselected sections of code written in the BASIC programming language.

Since Visual Basic is easy to learn and fast to write code with, it's sometimes used to prototype an application that will later be written in a more difficult but efficient language. Visual Basic is also widely used to write working programs.

2.3.2 Front End

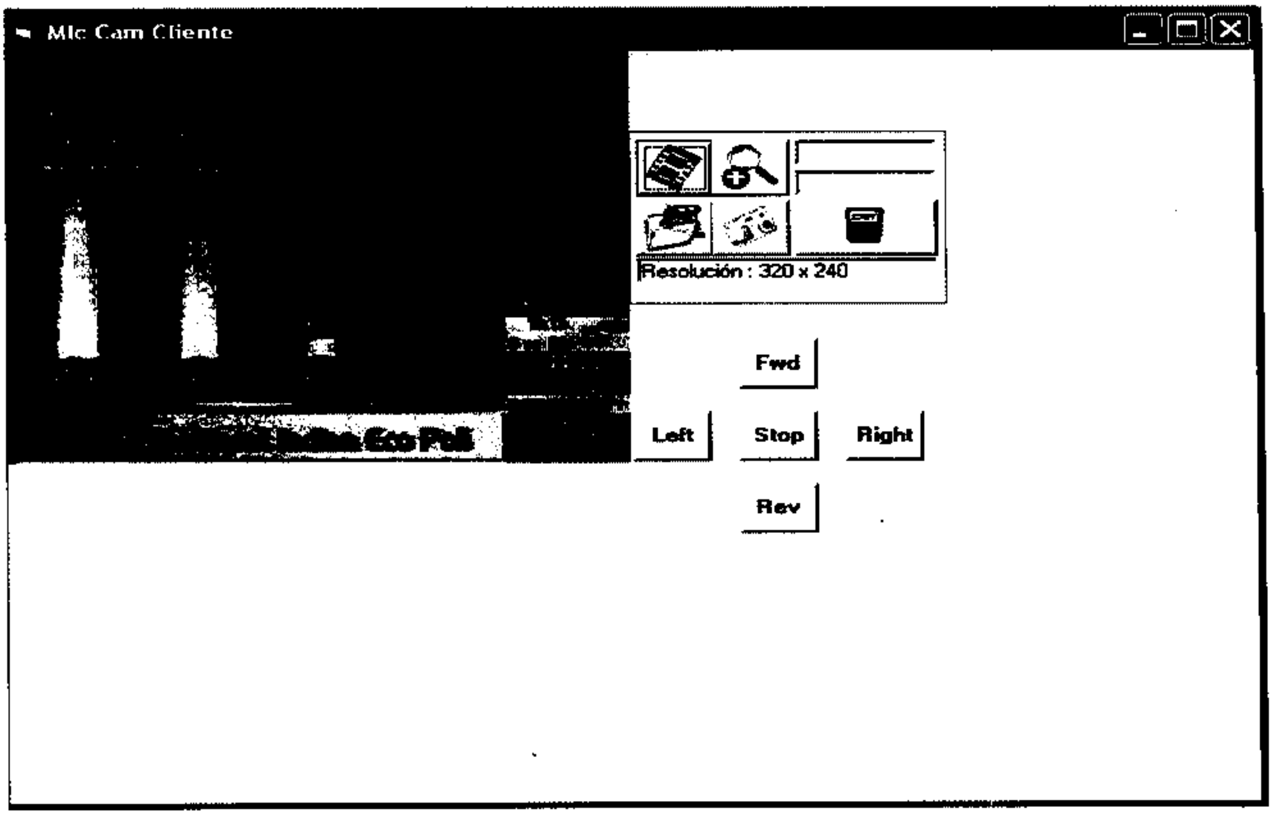


Fig 2.2 Font end

2.3.3 Coding for control action:

```
Public Declare Function Inp Lib "inpout32.dll" _

Alias "Inp32" (ByVal PortAddress As Integer) As Integer

Public Declare Sub Out Lib "inpout32.dll" _

Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)

Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Private Sub cmdforward_Click()

Out dataaddr, &HFE

End Sub

Private Sub cmdleft_Click()

Out dataaddr, &HF7

End Sub

Private Sub cmdreverse_Click()

Out dataaddr, &HFD

End Sub

Private Sub cmdright_Click()

Out dataaddr, &HFB

End Sub

Private Sub cmdstop_Click()
```

```
Out dataaddr, &HEF

End Sub
```

2.4 PARALLEL PORT INTERFACE

2.4.1 Introduction:

Actions performed from the user end can be transferred through parallel port by the execution of visual basic program along with dll file.

2.4.2Parallel port:

Computer transfer data in two ways: **parallel and serial.**

The parallel port is generally used to communicate with pc. It has a group of inputs and digital outputs, which can be used to make practical experimental of reading of data and control of devices. This work seeks to give more relevant aspects of the parallel port, which acts like an input output interface that works from a subordinate way to software routines.

It can be used as an alternative to the use of Programmable Logical Controllers (PLC) and Data Acquisition Cards.

2.4.3 Input Output Ports

The Input Output Ports constitute in the means for which the microprocessor of a computer communicates with its environment. Ports exist for each interaction of the unit of main prosecution with their auxiliary devices. This way, a port of entrance of the keyboard, a port exists of exit for the videotape, an input port for the mouse, etc. The Personal computer (PC) can address up to 64K ports of I/O. Each port is designated by a number. Next the addresses are listed in hexadecimal of the more usual port of I/O.

2.4.4 Port Layout:

The parallel port is formed by 17 lines of signs and 8 earth lines. The lines of signs are formed for three groups:

- 4 control lines
- 5 status lines
- 8 lines of data

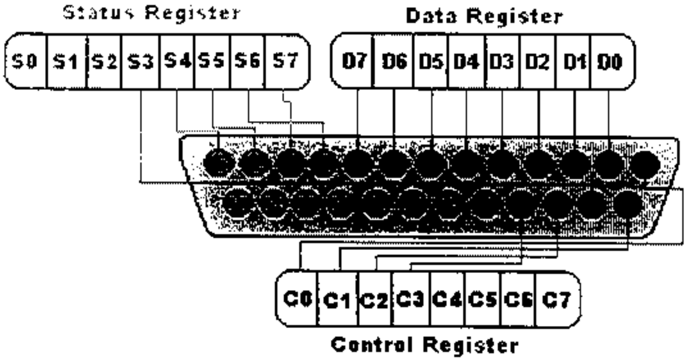


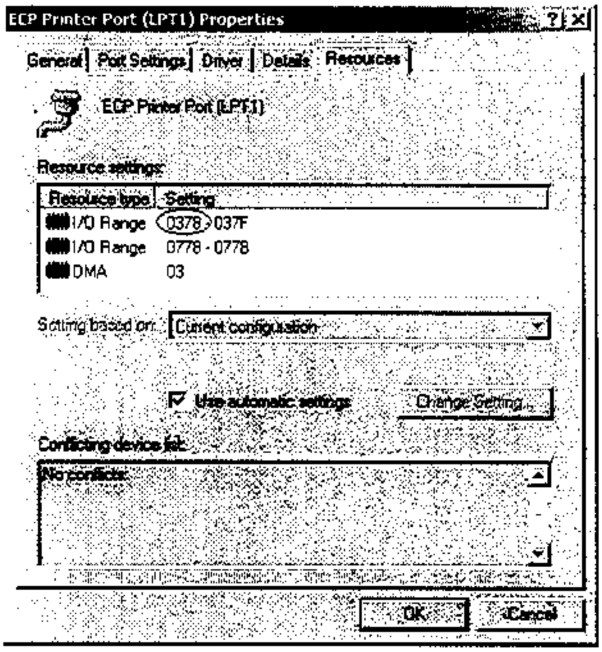
Fig 2.3 Parallel port layout

Status lines -exchange of messages, state indicators from the printer to the PC
Lines of data -gives the data of impression of the PC toward the printer and only in that address. Each one of these lines (control, state, data) it can be indexed in an independent way by means of a registration.

2.4.5 Determination of Base Address:

To write an interfacing program with parallel port or serial port, it is necessary to know the base address of the corresponding port in advance. And this can be confirmed by checking the Parallel Port properties in Device Manager. To do this, right click on My Computer, select Manage. Select Device Manager from the Computer Management console. Choose Parallel Port from the device tree, right click on it and select "Properties". Go to the "Resources" tab. Here the Base address of the parallel port is obtained. A sample parallel port properties window is shown below. The Base address is highlighted in circle.

Inpout32.dll, the device driver is embedded in the DLL and is installed and configured automatically at the very first call made to the library (DLL). This DLL file has to be copied to relevant directory (system directory or application directory) and exported functions are called for execution.



Resource settings:

Resource type	Setting
I/O Range	0378 - 037F
I/O Range	0778 - 0778
DMA	03

Setting based on: Current configuration

☑ Use automatic settings

Conflicting device list:

No conflicts.

Fig 2.4 Port properties window

Current Beta version of this library supports all the functions available in Inpout32.dll (Inp32() and Out32()), The library also contains functions Inpw32()/OutW32 which reads/writes 16 bit word and InpD32()/OutD32 which reads/writes 32 but DWords. Apart from I/O functions this library contains memory access functions also.

2.4.6 INTERFACING PARALLEL PORT WITH ENCODER

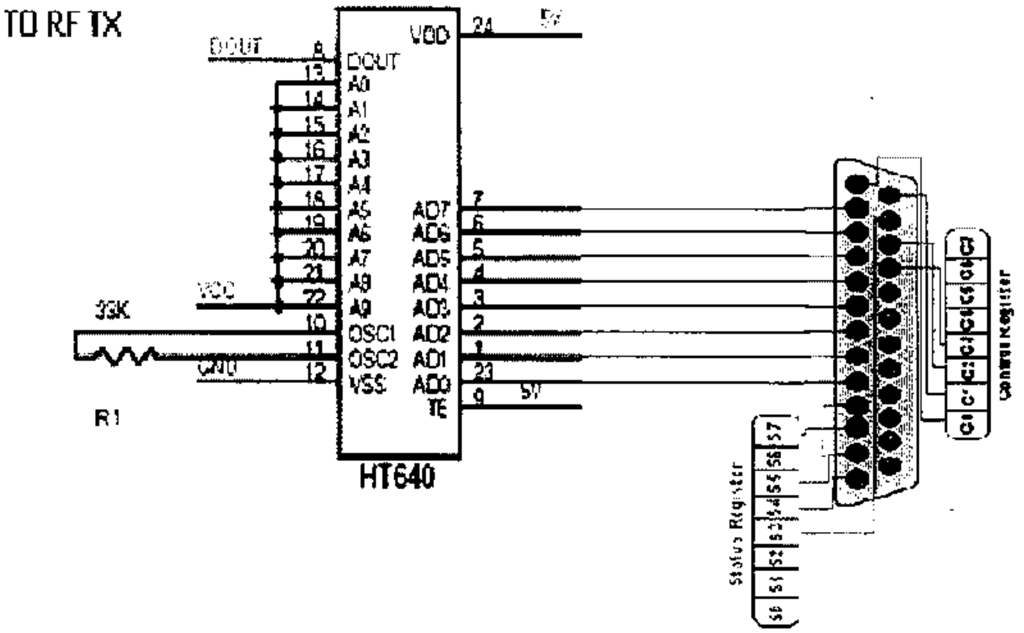


Fig 2.5 Interfacing parallel port

The output of the parallel port is given to the data pin of the encoder(HT640).using 8 data pins(AD0-AD7) 256 different operations can be performed.

The Hex combinations used in this project are

Hexadecimal code	Binary equivalent	operations
FE	1111 1110	FORWARD
FD	1111 1101	REVERSE
F7	1111 0111	LEFT
FB	1111 1011	RIGHT
EF	1110 1111	STOP

2.4.7 Dynamic link library

- A dynamic link library (DLL) is a collection of small programs, any of which can be called when needed by a larger program that is running in the computer. The small program that lets the larger program communicate with a specific device such as a printer or scanner is often packaged as a DLL program (usually referred to as a DLL file). DLL files that support specific device operation are known as device drivers.

The advantage of DLL files is that, because they don't get loaded into random access memory (RAM) together with the main program, space is saved in RAM. When and if a DLL file is needed, then it is loaded and run.

DLL files are dynamically linked with the program that uses them during program execution rather than being compiled with the main program. The dll file used for driving parallel port is **Inpout32.dll.**

Inpout32.dll

1) 'Inp32', reads data from a specified parallel port register.

2) 'Out32', writes data to specified parallel port register.

2.5 ENCODER AND TRANSMITTER

2.5.1 HT 640 Encoder:

The HT 640 is used as encoder. They are capable of encoding 18 bits of information which consists of N address bit and 18-N data bits. Each address/data input is externally programmable if bonded out. Various packages of the 3^{18} encoders offer flexible combination of programmable address/data is transmitted together with the header bits via an RF or an infrared transmission medium upon receipt of a trigger signal.

In HT 640 the input signal to be encoded is given to AD10-AD17 input pins of encoder. The input signal may be from key board, parallel port, microcontroller or any interfacing device. The encoder output address pins are shorted so the output encoded signal is the combination of (A0-A9) address signal and (AD10-AD17) data signal. The output encoded signal is taken from 8th which is connected to RF transmitter (TWS 434).

2.5.2 FEATURES:

- Operating voltage: 2V ~12V
- Low power and high noise immunity CMOS technology
- Low standby current
- Three words transmission
- Built in oscillator needs only 5% resistor
- Easy interface with an RF or infrared transmission media
- Minimal external components
- Operating frequency 433.90 MHz

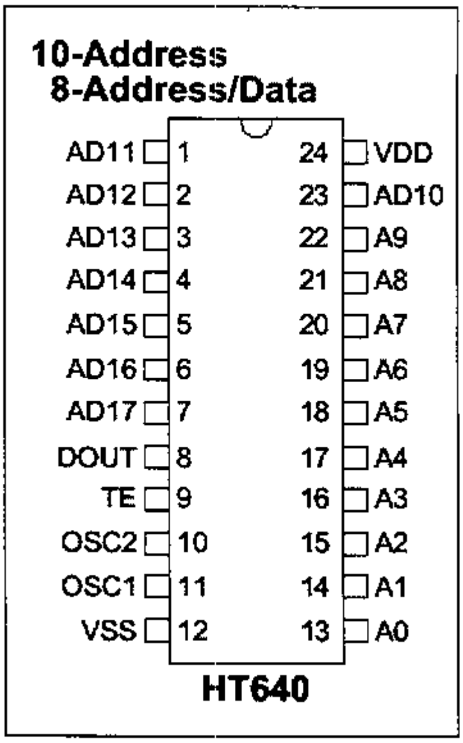


Fig 2.6 HT640 Encoder

2.5.3 RF TRANSMITTER:

Circuit Diagram:

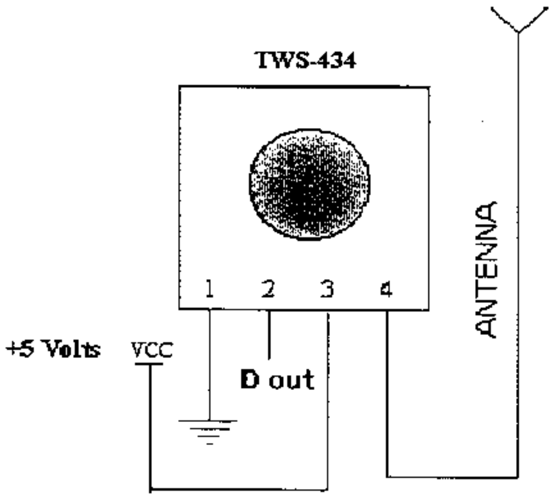


Fig 2.7 RF transmitter

The RF transmitter works under the concept of **ASK modulation**.

- It is driven by a single 9V supply from a battery
- It uses a common emitter amplifier biased with a voltage divider circuit
- The tank circuit is consists of L2 and C4 generating 433 MHz carrier signal.
- The parallel circuit components of 0.1mH inductor and 0.01uF capacitor make up the tank circuit.
- L and C are used as a filter
- The 0.01uF capacitor at the single entry point is an bypass

2.5.4 ENCODER WITH RF TRANSMITTER:

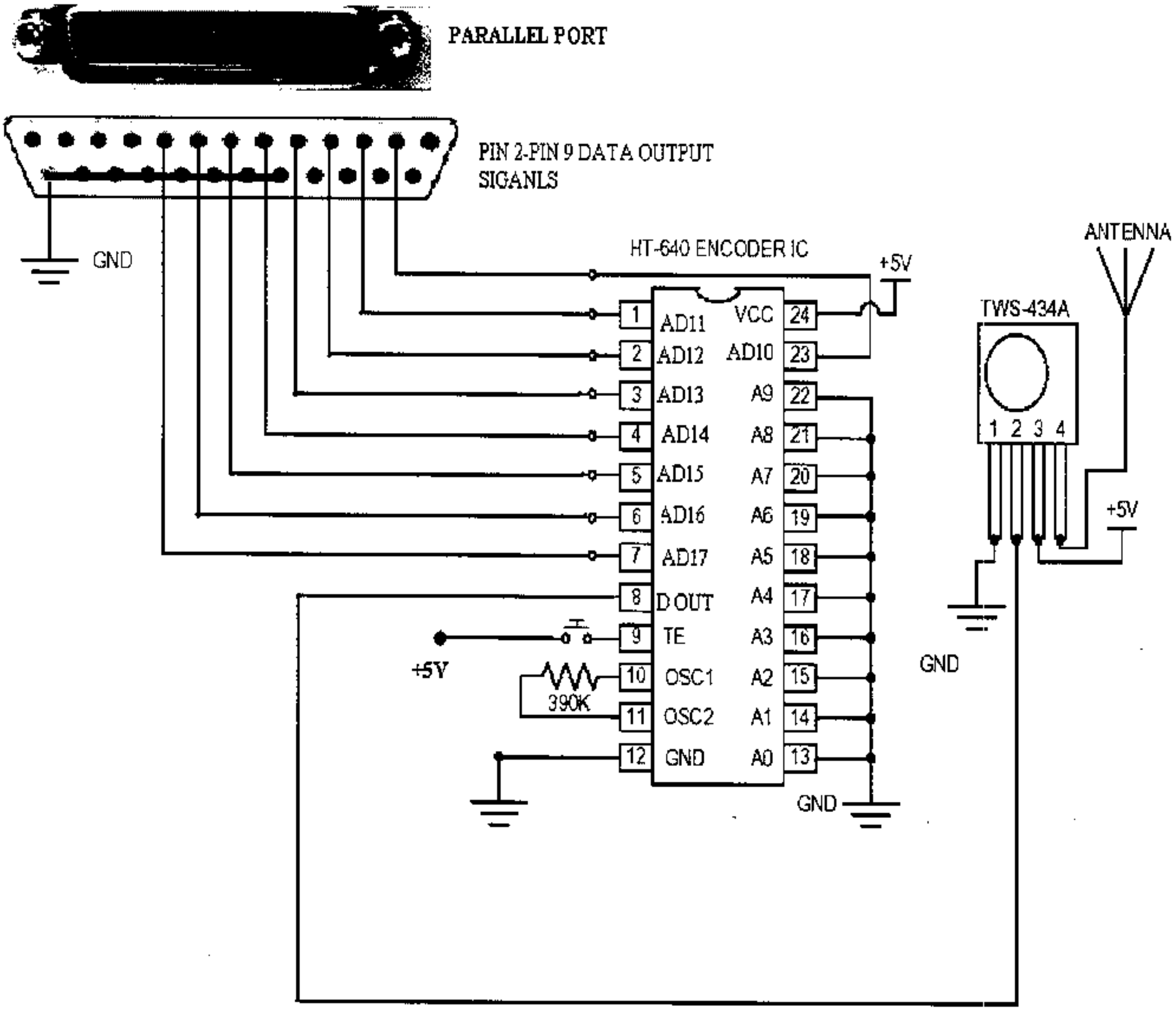


Fig 2.8 Transmitting section

3. VIDEO TRANSMISSION ROBOT

3.1 SYSTEM DESCRIPTION:

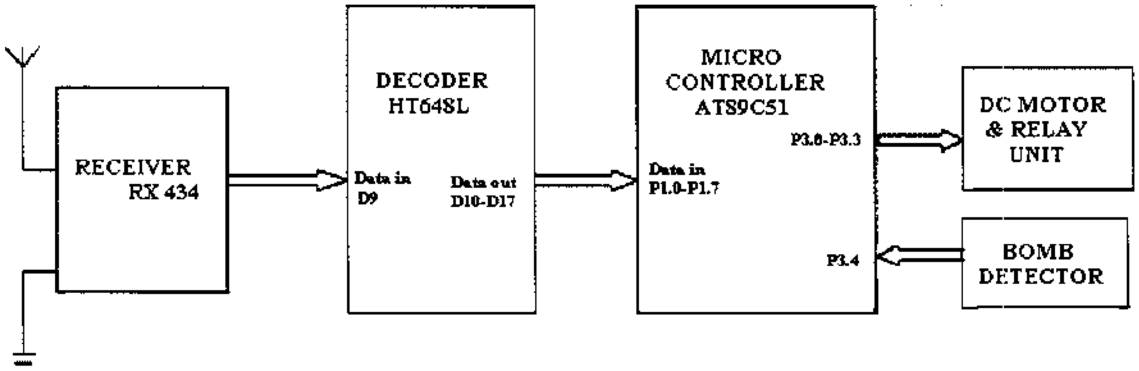


Fig 3.1 ROBOT SECTION

3.2 RECEIVER AND DECODER

3.2.1 HT 648L Decoder:

The HT648 is used as decoder. They are paired with 3^{18} series of encoders. The 3^{18} series of decoder receives serial address and data from that series of encoders that are transmitted by a carrier using an RF transmission medium. The VT pin also goes high to indicate a valid transmission.

The 3^{18} decoders are capable of decoding 18 bits of information that consists of N bits of address and 18-N bits of data. In this project the received encoded signal is 9th pin of the decoder. Now the decoder separate the address (A0-A9) and data signal (D0-D7). Then the output data signal is given to microcontroller or any other interfacing device.

3.2.2 FEATURES:

- Operating voltage: 2.4V~12V
- Low power and high noise immunity
- Low standby current
- Capable of decoding 18 bits of information
- 8~18 address pins
- 0~8 data pins
- Two times of receiving check
- Valid transmission indicator
- Minimal external components

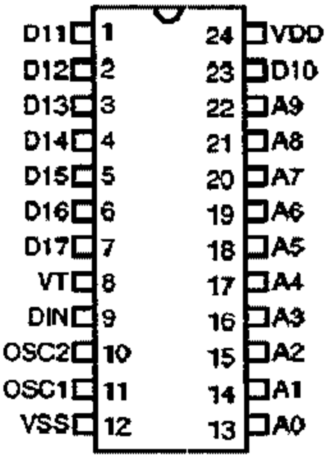
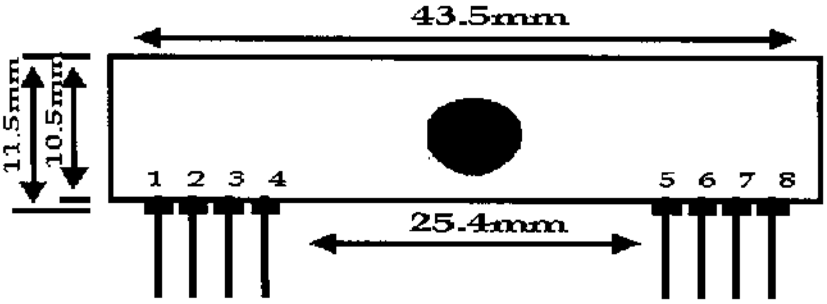


Fig 3.2 HT648L Decoder

3.2.3 RF RECEIVER:



pin 1 : Gnd
pin 2 : DigitaL Output
pin 3 : Linear Output
pin 4 : Vcc
pin 5 : Vcc
pin 6 : Gnd
pin 7 : Gnd
pin 8 :Ant (About 30 - 35 cm)

Fig 3.3 RF receiver

The RF receiver works under the concept of ASK modulation.

- It is driven by a single 9V supply from a battery
- It uses a common emitter amplifier biased with a voltage divider circuit
- The tank circuit is consists of L2 and C4 generating 433 MHz carrier signal.
- The parallel circuit components of 0.1mH inductor and 0.01uF capacitor make up the tank circuit.
- L and C are used as a filter
- The 0.01uF capacitor at the single entry point is an bypass

3.2.4 DECODER WITH RF RECEIVER

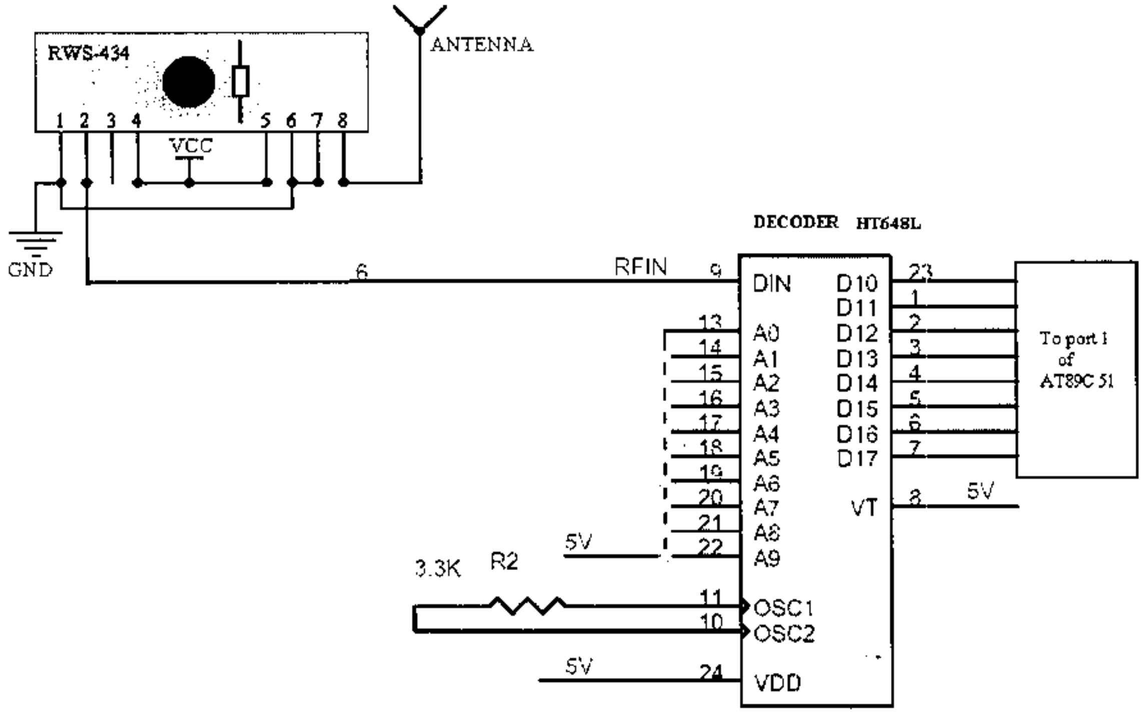


Fig 3.4 Decoder with RF receiver

3.3 MICRO CONTROLLER SECTION

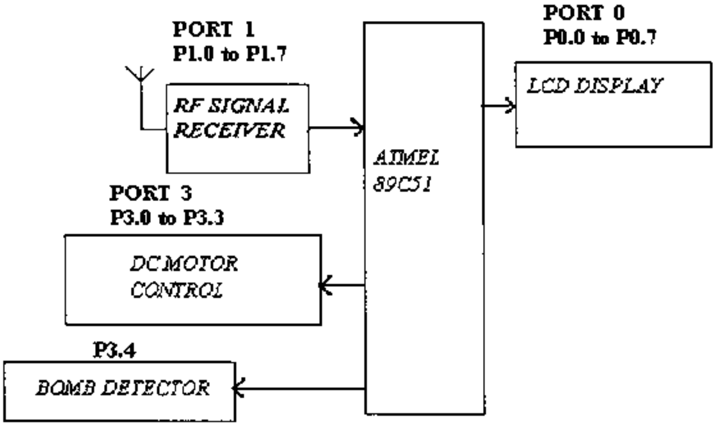


Fig 3.5 Microcontroller port details

Microcontrollers (MCUs) are intelligent electronic devices used inside robots. They deliver functions similar to those performed by a microprocessor (central processing unit, or CPU) inside a personal computer. MCUs are slower and can address less memory than CPUs, but are designed for real-world control problems. One of the major differences between CPUs and MCUs is the number of external components needed to operate them. MCUs can often run with zero external parts, and typically need only an external crystal or oscillator.

3.3.1 INTRODUCTION TO ATMEL MICROCONTROLLER

SERIES: 89C51 Family, **TECHNOLOGY:** CMOS

The major Features of 8-bit Micro controller **ATMEL 89C51**:

- 8 Bit CPU optimized for control applications
- Extensive Boolean processing (Single - bit Logic) Capabilities.
- On - Chip Flash Program Memory
- On - Chip Data RAM
- Bi-directional and Individually Addressable I/O Lines
- Multiple 16-Bit Timer/Counters
- Full Duplex UART
- Multiple Source / Vector / Priority Interrupt Structure
- On - Chip Oscillator and Clock circuitry.
- On - Chip EEPROM
- SPI Serial Bus Interface
- Watch Dog Timer

3.3.2 Functions of Microcontroller:

The microcontroller acts as the heart of the robot section. Various operations can be performed using microcontroller by suitably programming it. In this project we make use of microcontroller for performing

1. DC motor control
2. Bomb detection Control

3.3.2.1 DC MOTOR & RELAY UNIT

This circuit is designed to control the motor in the forward and reverse direction. It consists of two relays named as relay1, relay2. The relay ON and OFF is controlled by the pair of switching transistors. A Relay is nothing but electromagnetic switching device which consists of three pins. They are Common, Normally close (NC) and Normally open (NO). The common pin of two relay is connected to positive and negative terminal of motor through snubber circuit respectively. The relays are connected in the collector terminal of the transistors T2 and T4.

When high pulse signal is given to either base of the T1 or T3 transistors, the transistor is conducting and shorts the collector and emitter terminal and zero signals is given to base of the T2 or T4 transistor. So the relay is turned OFF state.

When low pulse is given to either base of transistor T1 or T3 transistor, the transistor is turned OFF. Now 12v is given to base of T2 or T4 transistor so the transistor is conducting and relay is turn ON. The NO and NC pins of two relays are interconnected so only one relay can be operated at a time.

The series combination of resistor and capacitor is called as snubber circuit. When the relay is turn ON and turn OFF continuously, the back emf may fault the relays. So the back emf is grounded through the snubber circuit.

Motor 1(placed in the left side)

- When relay 1 is in the ON state and relay 2 is in the OFF state, the motor is running in the forward direction.
- When relay 2 is in the ON state and relay 1 is in the OFF state, the motor is running in the reverse direction.

Motor 2(placed in the right side)

- When relay 3 is in the ON state and relay 4 is in the OFF state, the motor is running in the forward direction.
- When relay 3 is in the ON state and relay 4 is in the OFF state, the motor is running in the reverse direction.

Motor 1	**Motor 2**	**Direction of Robot**
Forward	Forward	Forward direction
Reverse	Reverse	Reverse direction
Reverse	Forward	Left direction
Forward	Reverse	Right direction
No Movement	No Movement	No action

3.3.2.2 CIRCUIT DIAGRAM OF DC MOTOR & RELAY UNIT:

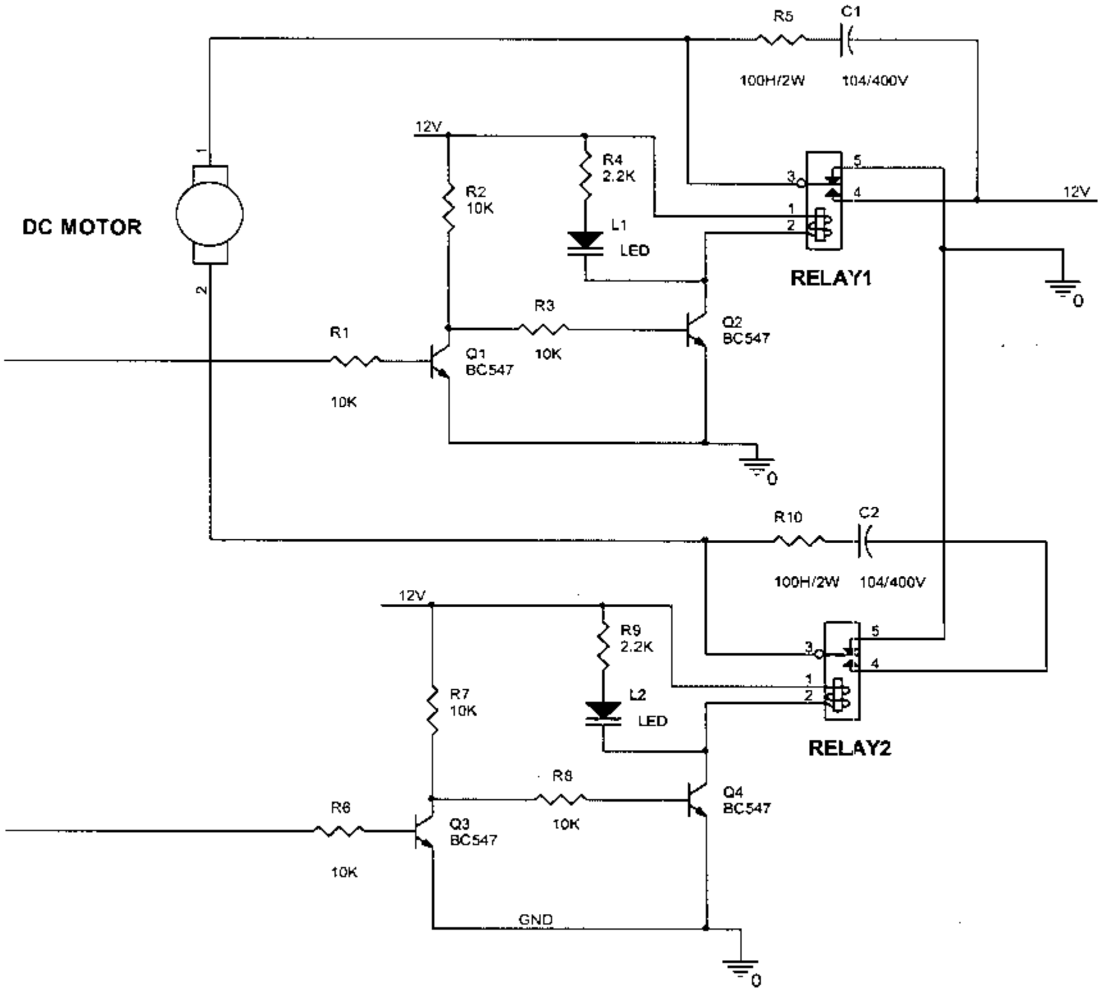


Fig 3.6 Dc Motor & Relay Circuit

Micro-controller with DC motor and Relay Unit

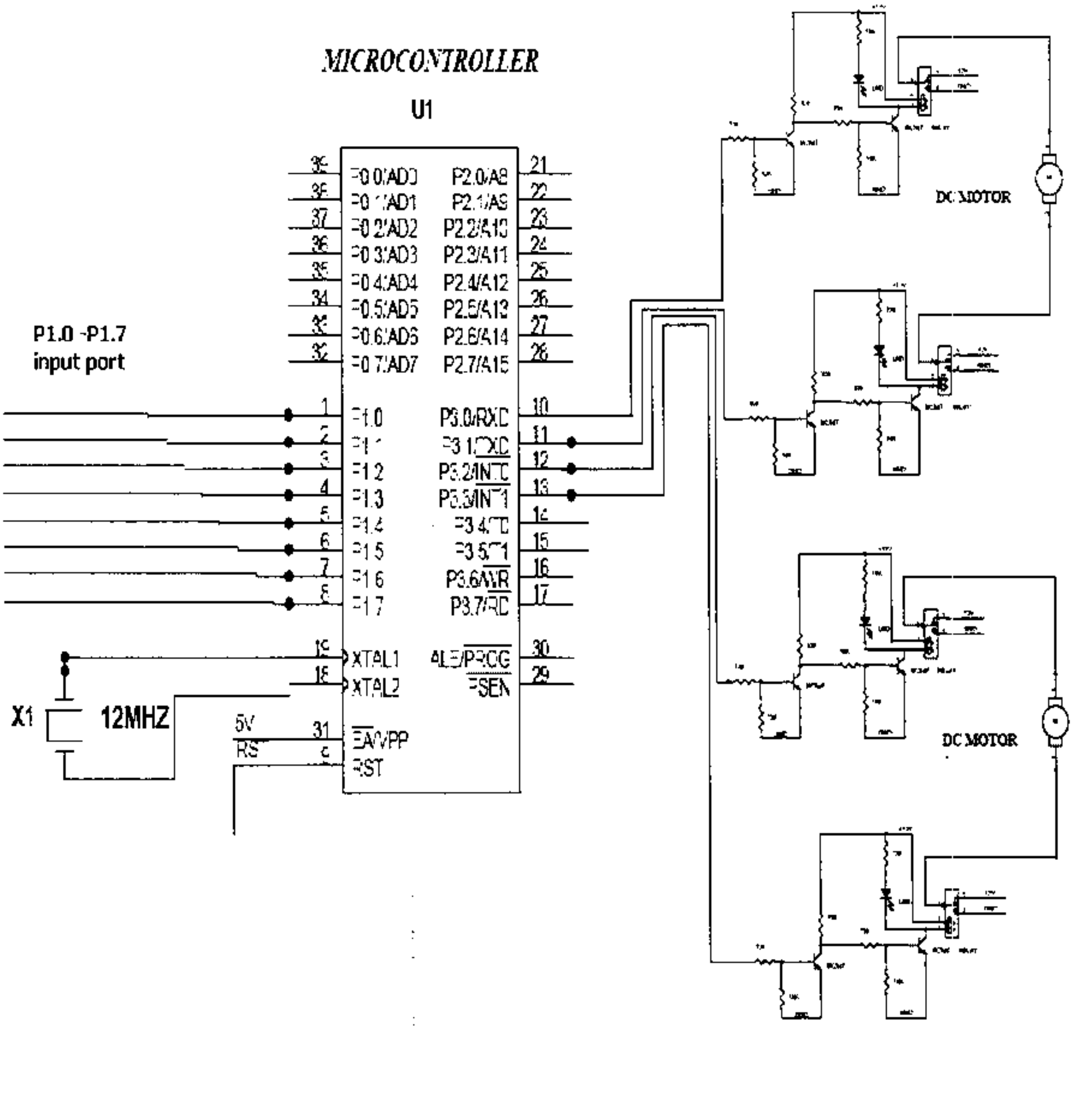


Fig 3.7 Microcontroller with relay unit

3.3.2.2 BOMB SENSING CIRCUIT :

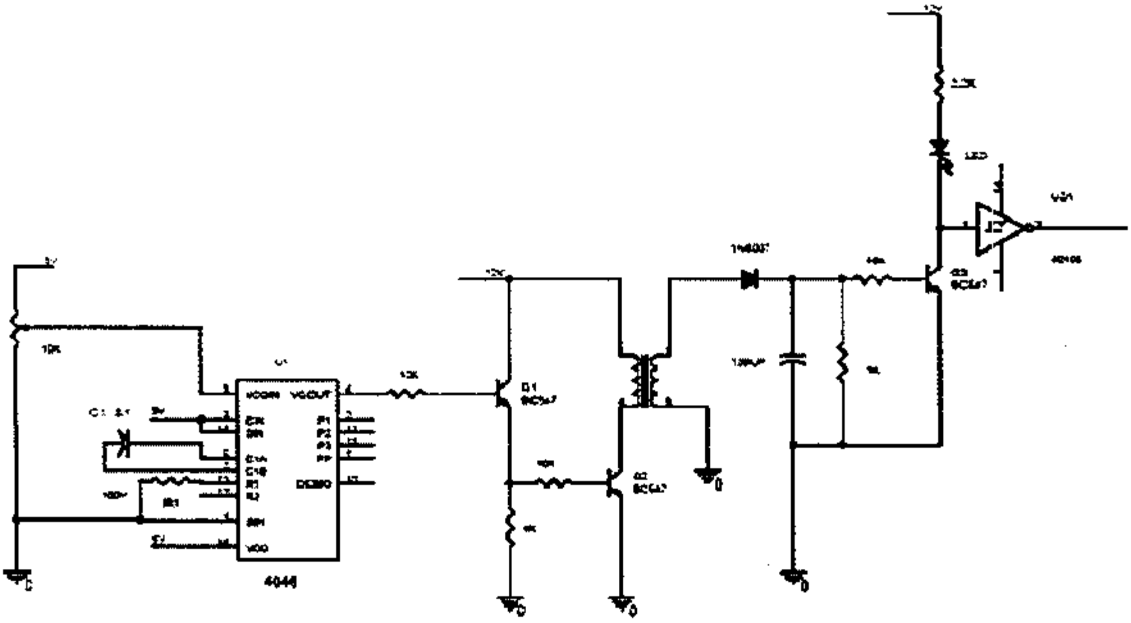


Fig 3.8 Metal Detector circuit

METAL DETECTOR:

Buzzer:

A **buzzer** or **beeper** is a signalling device, usually electronic, typically used in automobiles, household appliances such as a microwave oven, or game shows. It most commonly consists of a number of switches or sensors connected to a control unit that determines if and which button was pushed or a preset time has lapsed, and usually illuminates a light on the appropriate button or control panel, and sounds a warning in the form of a continuous or intermittent buzzing or beeping sound. Initially this device was based on an electromechanical system which was identical to an electric bell without the metal gong (which makes the ringing noise).

Often these units were anchored to a wall or ceiling and used the ceiling or wall as a sounding board. Another implementation with some AC-connected devices was to implement a circuit to make the AC current into a noise loud enough to drive a loudspeaker and hook this circuit up to a cheap 8-ohm speaker. Nowadays, it is more popular to use a ceramic-based piezoelectric sounder like a Sonalert which makes a high-pitched tone. Usually these were hooked up to "driver" circuits which varied the pitch of the sound or pulsed the sound on and off.

Circuit description:

The circuit is designed to control the buzzer. The buzzer ON and OFF is controlled by the pair of switching transistors (BC 547). The buzzer is connected in the Q2 transistor collector terminal.

When high pulse signal is given to base of the Q1 transistors, the transistor is conducting and close the collector and emitter terminal so zero signals is given to base of the Q2 transistor. Hence Q2 transistor and buzzer is turned OFF state.

When low pulse is given to base of transistor Q1 transistor, the transistor is turned OFF. Now 12v is given to base of Q2 transistor so the transistor is conducting and buzzer is energized and produces the sound signal.

BUZZER OPERATION

Voltage signal from microcontroller	Transistor Q1	Transistor Q2	Buzzer
1	on	off	off
0	off	on	cn

ALARM CIRCUIT

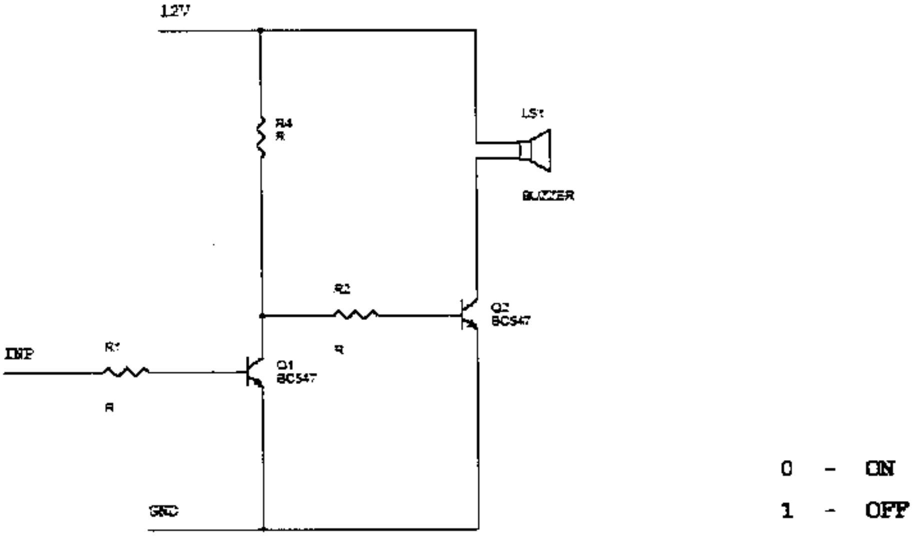


Fig 3.9 Alarm circuit

3.3.3 ATMEL 89C51 Pin configuration:

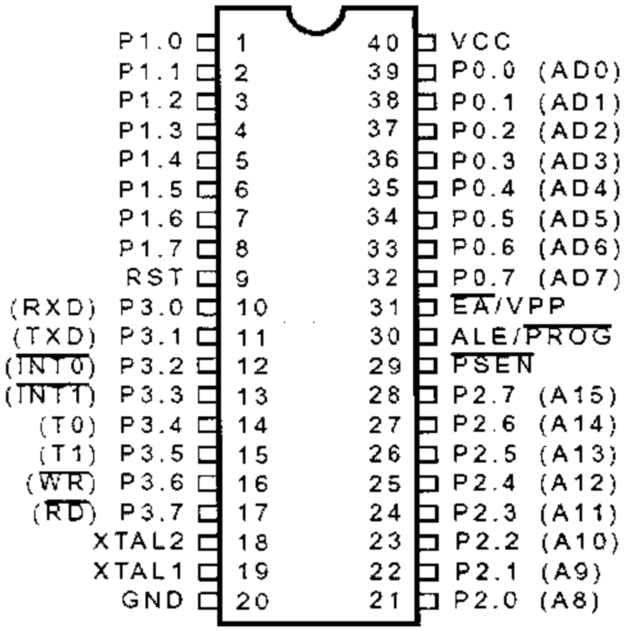

P1.0 1 40 VCC
P1.1 2 39 P0.0 (AD0)
P1.2 3 38 P0.1 (AD1)
P1.3 4 37 P0.2 (AD2)
P1.4 5 36 P0.3 (AD3)
P1.5 6 35 P0.4 (AD4)
P1.6 7 34 P0.5 (AD5)
P1.7 8 33 P0.6 (AD6)
RST 9 32 P0.7 (AD7)
(RXD) P3.0 10 31 $\overline{EA}$/VPP
(TXD) P3.1 11 30 ALE/$\overline{PROG}$
($\overline{INT0}$) P3.2 12 29 $\overline{PSEN}$
($\overline{INT1}$) P3.3 13 28 P2.7 (A15)
(T0) P3.4 14 27 P2.6 (A14)
(T1) P3.5 15 26 P2.5 (A13)
($\overline{WR}$) P3.6 16 25 P2.4 (A12)
($\overline{RD}$) P3.7 17 24 P2.3 (A11)
XTAL2 18 23 P2.2 (A10)
XTAL1 19 22 P2.1 (A9)
GND 20 21 P2.0 (A8)

Fig 3.10 ATMEL 89C51

3.4 MICROCONTROLLER PROGRAMMING

3.4 KEIL C51 C Compiler:

The Keil C51 C Compiler for the ATMEL 89 series microcontrollers provides more features than any other ATMEL 89 series C compiler available today.

The C51 Compiler allows you to write ATMEL 89 series microcontroller applications in C that, once compiled, have the efficiency and speed of assembly language. Language extensions in the C51 Compiler give you full access to all resources of the microcontroller.

The C51 Compiler translates C source files into relocatable object modules which contain full symbolic information for debugging with the μVision Debugger. The object file is then converted into Hex code by the compiler. In addition to these, the compiler generates a listing file which may optionally include symbol table and cross reference information.

3.4.1 Features

- Nine basic data types, including 32-bit IEEE floating-point,
- Flexible variable allocation with **bit**, **data**, **bdata**, **idata**, **xdata**, and **pdata** memory types,
- Interrupt functions may be written in C,
- Full use of the 8051 register banks,
- Complete symbol and type information for source-level debugging,
- Use of **AJMP** and **ACALL** instructions,
- Bit-addressable data objects,
- Built-in interface for the RTX51 Real-Time Kernel,
- Support for dual data pointers on Atmel, AMD, Cypress, Dallas Semiconductor, Infineon, Philips, and Transcend microcontrollers,
- Support for the Philips 8*x*C750, 8*x*C751, and 8*x*C752 limited instruction sets,
- Support for the Infineon 80C517 arithmetic unit.

3.4.2 FLASH 89 PROGRAMMER:

Specification:

Power Supply	:	14-18V DC or 12-16V AC
Interface	:	USB and RS-232, 9-pin D connector
Data Speed	:	57600 bps, 8 bits, no parity, 1 stop, no flow control
File format	:	Intel 8-bit HEX
Program Sockets	:	40 pin DIP - 0.6" & 20 pin DIP 0.3" ZIF socket
Software	:	Works on Windows 95, 98, Me, 2000, NT, XP

Flash 89 programmer is used to burn the hexadecimal code obtained from the KEIL C51 software into the ATMEL microcontroller by using ATMEL parallel programmer kit.It is a powerful flash microcontroller programmer for the Atmel 89 series.

Following are the main features of this software,

- Read and write the Intel Hex file.
- Chip Erase.
- Verify.
- Lock.
- Read Device Signature.

3.4.2.2 SUPPORTED DEVICES:

This programmer software presently supports the following devices

AT89C51	AT89S51	AT89C1051U	D87C51
AT89C52	AT89S52	AT89C2051	D87C52
AT89C55	AT89S53	AT89C4051	
AT89C55WD	AT89S8252	AT89C51RC	

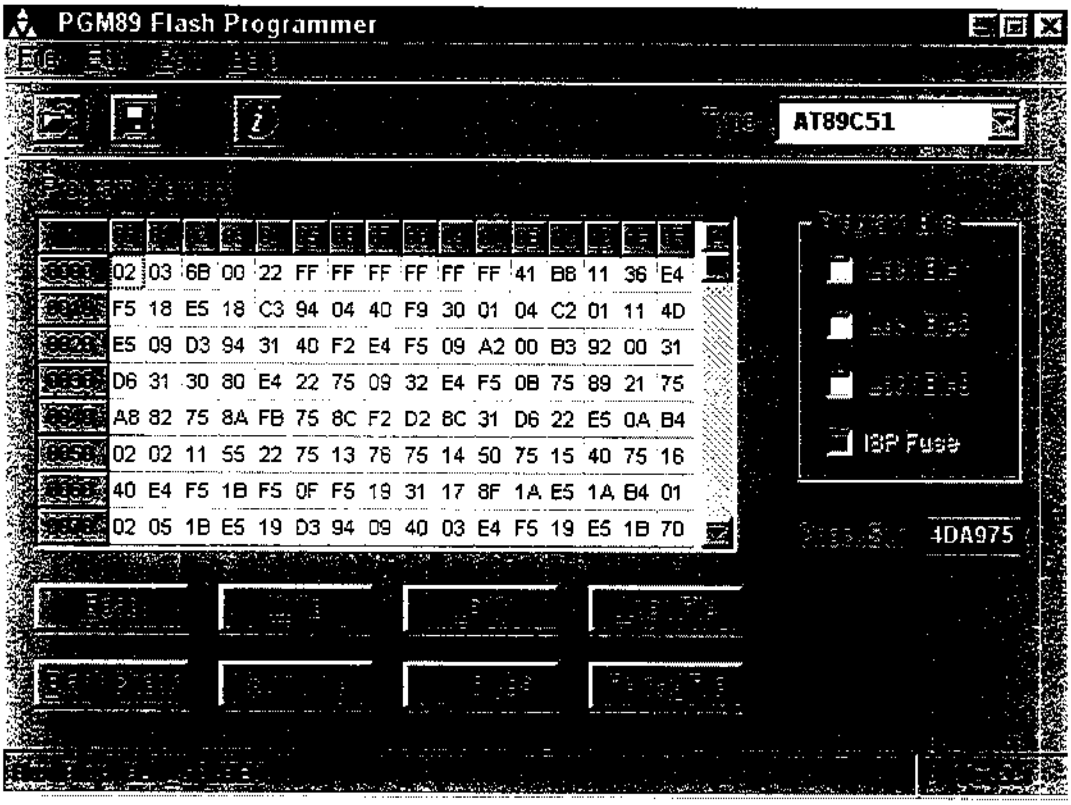
This programmer has intelligent onboard firmware and connects to the serial port. It can be used with any type of computer and requires no special hardware. The programmer is connected to the PC through serial communication port. The programmer connects to a host computer using a standard RS232 serial port. Flash 89 Programmer comes with window based software for easy programming of the devices

Flash 89 programmer is a PLUG and PLAY tool. All devices have signature bytes that the programmer reads to automatically identify the chip. There's no need to select the device type. All devices also have a number of **lock bits** to provide various levels of software and programming **protection**.

These lock bits are fully programmable using this programmer.Lock bits are useful to protect the program to be read back from microcontroller only allowing erase to reprogram the microcontroller.

3.4.2.3 PROGRAMMER SCREEN

The Flash 89 Programmer screen used to load the hex code into microcontroller is shown below with a sample hex code. The operations mentioned above can be done using this screen. We can also enable the Lock Bits for protection.



```
02 03 6B 00 22 FF FF FF FF FF FF 41 B8 11 36 E4
F5 18 E5 18 C3 94 04 40 F9 30 01 04 C2 01 11 4D
E5 09 D3 94 31 40 F2 E4 F5 09 A2 00 B3 92 00 31
D6 31 30 80 E4 22 75 09 32 E4 F5 0B 75 89 21 75
A8 82 75 8A FB 75 8C F2 D2 8C 31 D6 22 E5 0A B4
02 02 11 55 22 75 13 76 75 14 50 75 15 40 75 16
40 E4 F5 1B F5 0F F5 19 31 17 8F 1A E5 1A B4 01
02 05 1B E5 19 D3 94 09 40 03 E4 F5 19 E5 1B 70
```

Fig 3.11 PGM89 Programmer screen

3.4.3 BURNING PROCESS

Burning is nothing but the process of loading the program into the microcontroller using suitable software. The programmer is connected to the PC through RS232 serial communication port. The process is given below

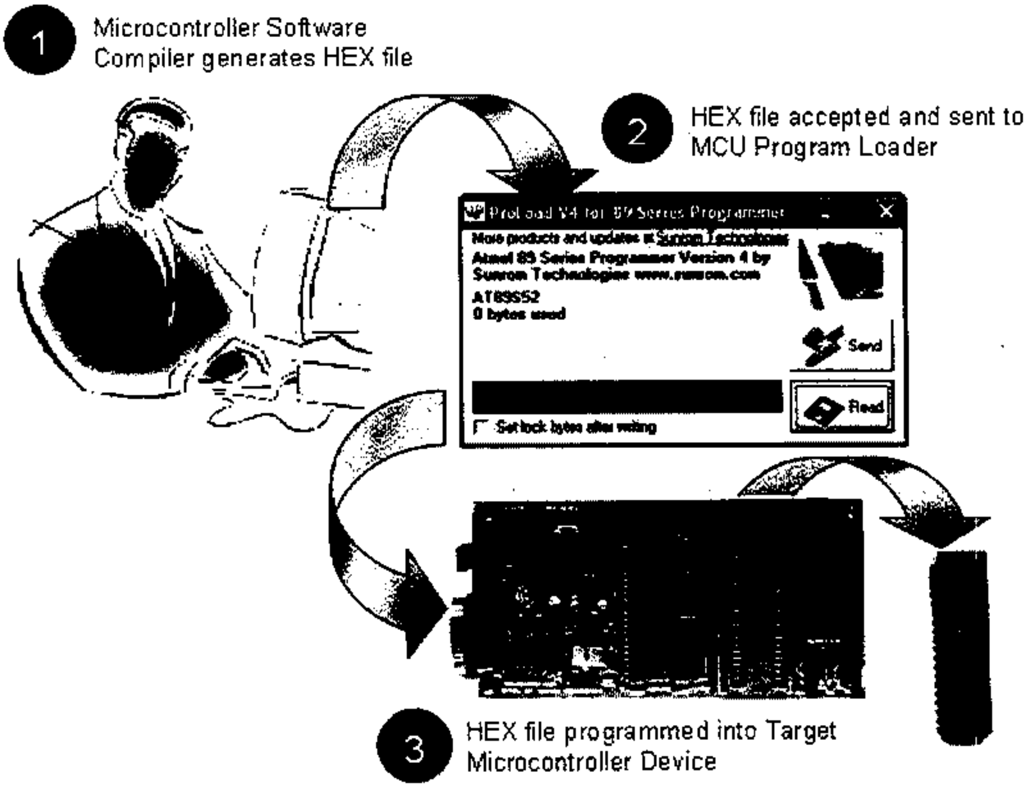
1. The Hexadecimal Code required for the microcontroller is generated using the KEIL software.

2. Using the Flash 89 Programmer the Hex file is sent to the MCU Program loader via the serial port of the PC.

3. The Hex file is then transferred into the target microcontroller device.

By this way we can program the Hex code into microcontroller. The diagrammatic representation is shown below.

By this way the microcontroller can be programmed.



1 Microcontroller Software Compiler generates HEX file

2 HEX file accepted and sent to MCU Program Loader

ProLoad V4 for 89 Series Programmer

More products and updates at Sunrom Technologies

Atmel 89 Series Programmer Version 4 by Sunrom Technologies www.sunrom.com

AT89S52

0 bytes used

□ Set lock bytes after writing

3 HEX file programmed into Target Microcontroller Device

3.4.5 FLOWCHART:

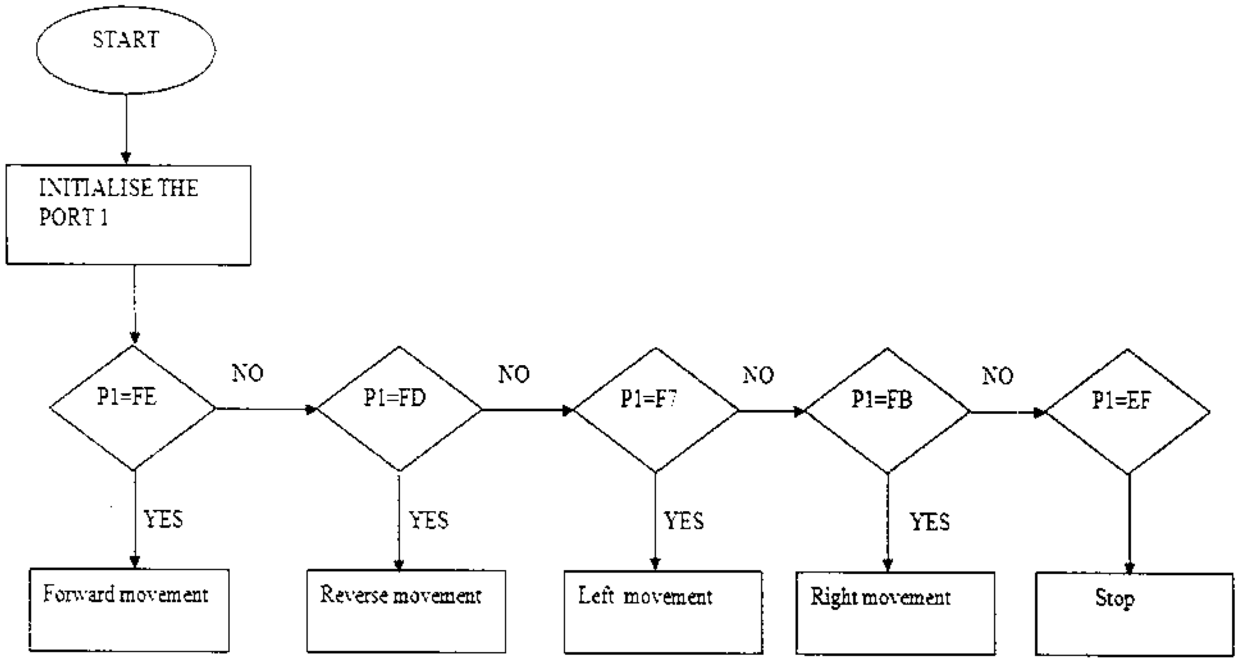


Fig 3.12 Flowchart For DC motor Control

3.4.6 SOFTWARE DESCRIPTION:

DC MOTOR CONTROL:

```
#include <AT89X51.H>

sbit lef_rev  =P3^0;                                    //PORT 3 as output port

sbit lef_for  =P3^1;

sbit rit_for  =P3^2;

sbit rit_rev  =P3^3;

sbit alarm    =P3^5;

sbit sen      =P3^6;

void forward();

void revers();

void lef_t();

void righ_t();

void stp();

void delay(unsigned int);


void main()

{

 sen=0;
```

```
        while(1)
         {
          if(P1==0XFE) forward();                              //PORT 1 as input port
          if(P1==0XFD) revers();
          if(P1==0XFB) lef_t();
          if(P1==0XF7) righ_t();
          if(P1==0XEF) stp();
          if(sen==1) { stp();alarm=0; delay(65000); delay(45000); alarm=1;}
         }
}
void revers()                                                  //Moves in the Reverse direction
{
 rit_rev=lef_rev=1;//on
 rit_for=lef_for=0;//off
}
void forward()                                                  //Moves in the Forward direction
{

rit_for=lef_for=1;//on
 rit_rev=lef_rev=0;//off
}
```

```
}


void righ_t()                                              //Moves in the Right direction
{
 rit_for=lef_rev=1; //on
 rit_rev=lef_for=0;//off
}


void lef_t()                                                //Moves in the Left direction
{
 rit_rev=lef_for=1;//off
 rit_for=lef_rev=0; //on
}
void stp()                                                 //Halts  the Robot
{
 rit_for=lef_for=1; //off
 rit_rev=lef_rev=1; //off
}
void delay(unsigned int ss)
{
 while(ss--);
}
```

3.5 VIDEO TRANMITTER SECTION

This section provides vision to the telerobot. One of the most useful gadgets a video enthusiast can have is a low-power TV transmitter. Such a device can transmit a signal from a camera to any TV in a home or system. When connected to a video camera, a TV transmitter can be used in surveillance for monitoring a particular location.

The TV Transmitter combines line level audio and video signals, and transmits the resulting signal up to 300 feet. The circuit can be powered from a 9-volt battery. It is suggested that a 12-volt DC supply during be used during the alignment procedure. This would ensure maximum transmission range and best possible picture. Aligning the TV Transmitter is a very simple procedure. The Transmitter's output can be tuned to be received on any TV channel by tuning into the particular frequency.

3.5.1 Circuit Description

The description of the video transmitter circuit is as follows. Video signals input at jack J1 are first terminated by resistor R6 and coupled through capacitor C1 to clamping-diode D1. The clamping forces the synchronous pulses to a fixed DC level to reduce blooming effects. Potentiometer R3 is used to set the gain of the video signal; its effect is similar to that of the contrast control on a TV set. Bias-control R7 can be used to adjust the black level of the picture so that some level of signal is transmitted, even for a totally dark picture. That way, a TV receiver can maintain proper synchronous. Potentiometers R3 and R7 are cross adjusted for the best all-around performance. RF-transformer T1 and its internal capacitor form the tank circuit of a Hartley oscillator that's tuned to 4.5 megahertz. Audio signals input at J2 are coupled to the base of Q3 via C2 and R4: the audio signal modulates the base signal of Q3 to form an audio subcarrier that's 4.5 MHz higher than the video-carrier frequency. The FM modulated subcarrier is applied to the modulator section through C5 and R9. Resistor R9 adjusts the level of the subcarrier with respect to the video signal. Transistors Q1 and Q2 amplitude modulate the video and audio signals onto an RF-carrier signal. The operating frequency is set by coil L4, which is 3.5 turns of 24- gauge enameled wire on a form containing a standard ferrite slug. That coil is

part of a Colpitts tank circuit also containing C7 and C9. The tank circuit forms Q4's feedback network, so Q4 oscillates at the set frequency The RF output from the oscillator section is amplified by Q5 and Q6, whose supply voltage comes from the modulator section. Antenna matching and low-pass filtering is performed by C12, C13, and L1. Resistor R12 is optional; it is added to help match the output signal to any kind of antenna.

3.5.2 Circuit Diagram:

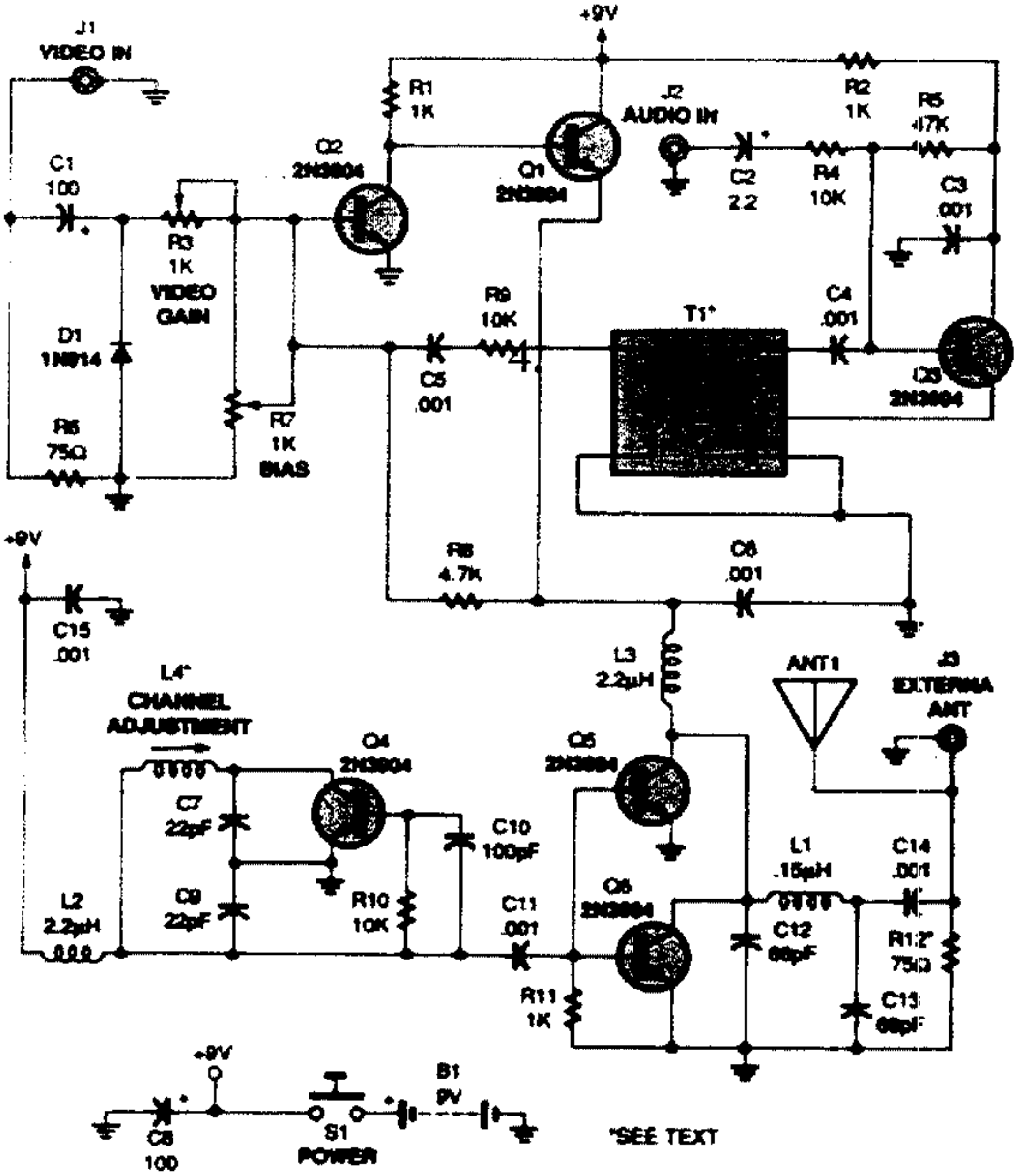


Fig 3.12 Video transmitting circuit

4. VIDEO RECEPTION IN PC

4.1 SYSTEM DESCRIPTION OF VIDEO RECEPTION

4.1.1 Hardware Unit:

The video signals transmitted by the robot section are captured using a video receiver circuit. The hardware block diagram of the video reception circuit is shown below.

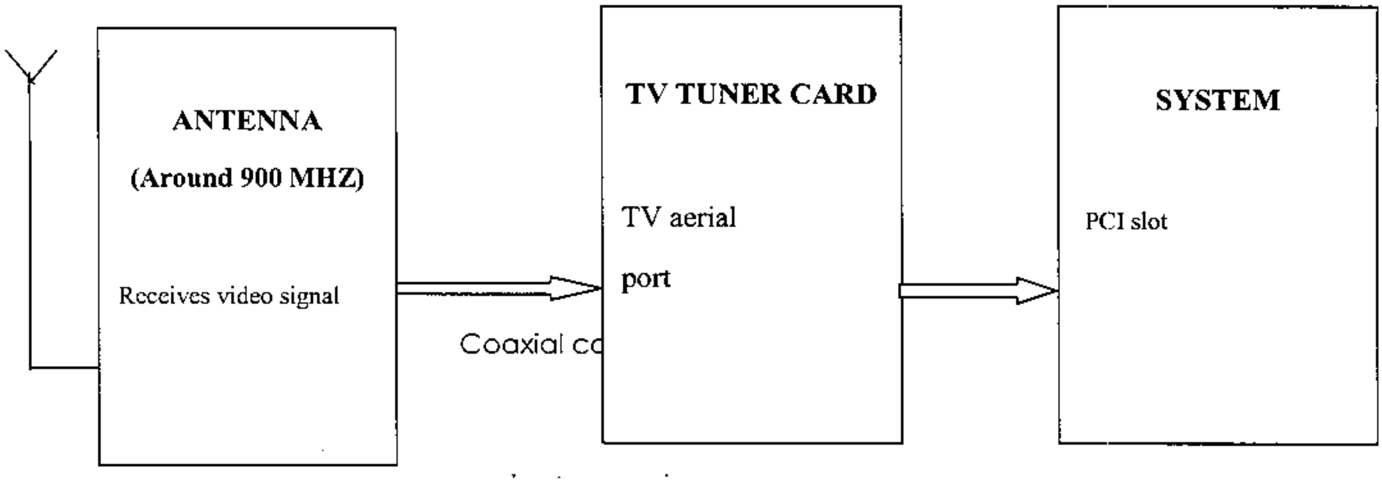


Fig 4.1 Video reception unit

The block diagram shows that the hardware unit has three main parts

- Antenna,
- TV Tuner card and
- VB screen.

The video signals transmitted in a frequency of around 900 MHz are captured using the antenna attached with a co – axial cable. The video signals are then sent to the TV tuner card via its TV aerial port through co- axial cable. Then by tuning for the required frequency i.e. around 900 MHz using the using the tuner card the captured video is seen. For viewing the video in Visual Basic screen we make use of coding which gets the video from source - tuner card and displays it in VB Screen.

4.2 INTERFACING TV TUNER CARD WITH PC:

A TV tuner card is a computer component that allows television signals to be received by a computer. Most TV tuners also function as video capture cards , allowing them to view the signals captured by antenna in computer. The TV tuner card is shown below.

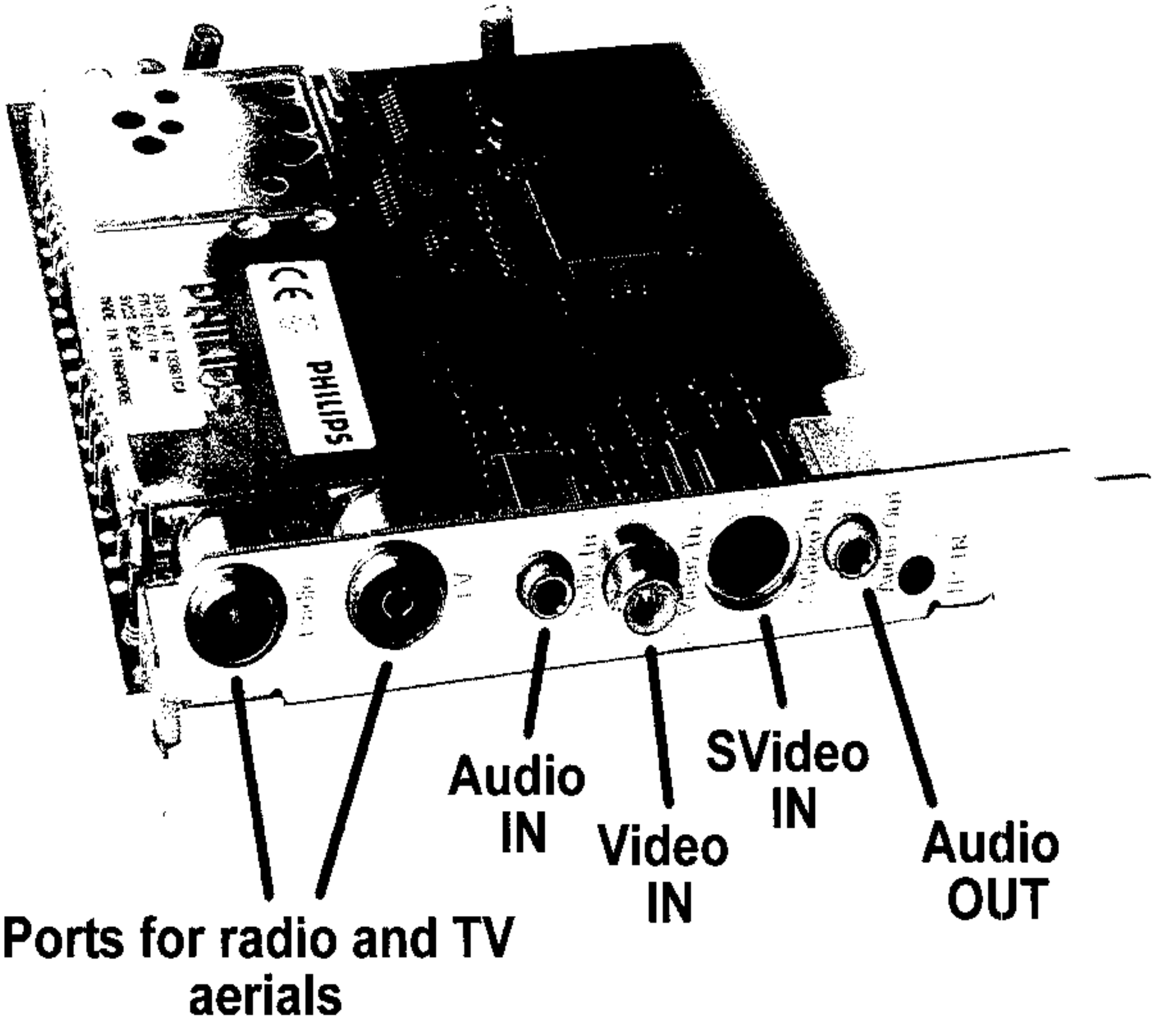


Fig 4.2 Tuner card

The tuner card consists of several ports such as

- Input for TV aerials from antenna
- Input for Video & SVideo
- Input for audio signals
- Audio input.

The input from antenna is given to the TV video aerial port. Whatever frequency the input be on, it is converted into a pre-determined intermediate frequency (IF) (IF contains both audio and video information).This "pre-determined" frequency varies depending on each TV system (eg: PAL; SECAM, NTSC, etc.) has a unique IF. Whatever the IF is, the tuner takes in all the millions of possible frequencies of radio waves in the universe, and filters out just the right frequency for display.

The two main works done by a tuner card are

- Video processing
- Audio Processing

4.2.1 Video Processing:

The IF which comes from the tuner module, needs to be decoded, and transformed into a viewable format. This is the job of the Video Processor. The viewable format may be like bitmap format, RGB format, YUV format, VGA format. The video signals that is seen on Computer Screen, is nothing but Digitized Video Data from the Video Processor being displayed by VGA adapter. Video Processing is done in into two steps as

1. Video Processor Digitizes Video Data and dumps it into the "frame buffer".
2. VGA adapter fetches Video data from the frame buffer, and displays it on screen.

4.2.2 Audio Processing:

Tuner Cards typically handle sound in two different ways.

The first method uses thc audio processor to demodulate sound from the IF .The audio signal thus obtained is routed to an external audio jack, from where it is re-routed to the line input of a separate sound card by means of a suitable external cable. The first method can avail of that luxury only by talking to the sound driver of the separate sound card. The advantage of this method is we can take the audio to sound driver of the separate sound card.

The second approach is for the audio processor to demodulate sound from the IF, convert it into Digital Samples, and use DMA technique to move these Samples to the sound card via the internal system bus (eg: The PCI bus), and from there, to use the sound card to reconvert the digital samples back to the audio signal. This method is more complicated, but more flexible, as the video sound levels are controllable on the tuner card itself.

4.3 CODING FOR VIDEO RECEPTION

VB coding:

```
Option Explicit

Dim Terminado As Boolean

Dim dataaddr

Public Sub Enviar(Linea As String)

    If Winsock1.State = sckConnected Then Winsock1.SendData Linea: DoEvents Else Bsent = False

End Sub

Public Sub SaveJPG(ByVal pict As StdPicture, ByVal filename As String, quality As Byte)

Dim tSI As GdiplusStartupInput

Dim lRes As Long

Dim lGDIP As Long

Dim lBitmap As Long


  ' Initialize GDI+

  tSI.GdiplusVersion = 1

  lRes = GdiplusStartup(lGDIP, tSI)

    If lRes = 0 Then

    ' Create the GDI+ bitmap
' from the image handlc
   lRcs = GdipCreateBitmapFromHBITMAP(pict.Handle, 0, lBitmap)
     If lRes = 0 Then
     Dim tJpgEncoder As GUID
     Dim tParams As EncoderParameters
      ' Initialize the encoder GUID
     CLSIDFromString StrPtr("{557CF401-1A04-11D3-9A73-0000F81EF32E}"), _
              tJpgEncoder
        ' Initialize the encoder parameters
     tParams.Count = 1
     With tParams.Parameter ' Quality
       ' Set the Quality GUID
       CLSIDFromString StrPtr("{1D5BE4B5-FA4A-452D-9CDD-5DB35105E7EB}"), .GUID
       .NumberOfValues = 1
       .type = 4
       .Value = VarPtr(quality)
     End With
          ' Save the image
     lRes = GdipSaveImageToFile( _
          lBitmap, _
   StrPtr(filename), _
              tJpgEncoder, _
              tParams)

        ' Destroy the bitmap
        GdipDisposeImage lBitmap
              End If
          ' Shutdown GDI+
     GdiplusShutdown lGDIP
   End If
End Sub
Private Sub CmdCompresion_Click()
'  /*
'   * Display the Compression dialog when "Compression" is selected from
'   * the menu bar.
'   */
    capDlgVideoCompression lwndC
End Sub
End Sub
Private Sub CmdFotos_Click()
```

```
Static Contador
 Dim sFile As String * 250
 Dim lSize As Long

'// Setup swap file for capture
 lSize = 1000000
 sFile = App.Path & "\" & Contador & Format(Now, "hhmmss") & ".dib"
 Call SendMessageS(lwndC, WM_CAP_FILE_SAVEDIB, 0, sFile)
  Contador = Contador + 1
    Dim iresult As Long ' no controlo si esta en c:\
    If Dir(App.Path & "\camara.wav") <> "" Then
    iresult = mciExecute("Play " & App.Path & "\camara.wav")
    End If
End Sub
Private Sub CmdMas_Click()
' /*
'  * Display the Video Format dialog when "Format" is selected from the
'  * menu bar.
'  */
    capDlgVideoFormat lwndC
```

```
ResizeCaptureWindow lwndC

End Sub

Private Sub CmdPropiedades_Click()

'  /*

'   * Display the Video Source dialog when "Source" is selected from the

'   * menu bar.

'   */

      capDlgVideoSource lwndC

End Sub

Private Sub CmdVideo_Click()

' /*

'  * If Start is selected from the menu, start Streaming capture.

'  * The streaming capture is terminated when the Escape key is pressed

'  */

     Dim sFileName As String

    Dim CAP_PARAMS As CAPTUREPARMS

    Static Contador

     capCaptureGetSetup lwndC, VarPtr(CAP_PARAMS), Len(CAP_PARAMS)

     CAP_PARAMS.dwRequestMicroSecPerFrame = (1 * (10 ^ 6)) / 30  ' 30 Frames per second

    CAP_PARAMS.fMakeUserHitOKToCapture = True
```

```
  CAP_PARAMS.fCaptureAudio = False
    capCaptureSetSetup lwndC, VarPtr(CAP_PARAMS), Len(CAP_PARAMS)
   sFileName = App.Path & "\" & Contador & "myvideo.avi"
    Contador = Contador + 1
   capCaptureSequence lwndC  ' Start Capturing!
   capFileSaveAs lwndC, sFileName  ' Copy video from swap file into a real file.
End Sub
Private Sub Form_Load()
dataaddr = &H378
   Dim lpszName As String * 100
   Dim lpszVer As String * 100
   Dim Caps As CAPDRIVERCAPS
   Calidad = 75
     '//Create Capture Window
   capGetDriverDescriptionA 0, lpszName, 100, lpszVer, 100  '// Retrieves driver info
   lwndC = capCreateCaptureWindowA(lpszName, WS_VISIBLE Or WS_CHILD, 0, 0, 160, 120, Me.hWnd, 0)
     '// Set title of window to name of driver
   SetWindowText lwndC, lpszName
     '// Set the video stream callback function
```

```
capSetCallbackOnStatus lwndC, AddressOf MyStatusCallback

capSetCallbackOnError lwndC, AddressOf MyErrorCallback

  '// Connect the capture window to the driver

If capDriverConnect(lwndC, 0) Then

   '/////

   '// Only do the following if the connect was successful.

   '// if it fails, the error will be reported in the call

   '// back function.

   '/////

   '// Get the capabilities of the capture driver

   capDriverGetCaps lwndC, VarPtr(Caps), Len(Caps)

    '// If the capture driver does not support a dialog, grey it out

   '// in the menu bar.

'     If Caps.fHasDlgVideoSource = 0 Then mnuSource.Enabled = False

'     If Caps.fHasDlgVideoFormat = 0 Then mnuFormat.Enabled = False

'     If Caps.fHasDlgVideoDisplay = 0 Then mnuDisplay.Enabled = False


   '// Turn Scale on

   capPreviewScale lwndC, True

    '// Set the preview rate in milliseconds

   capPreviewRate lwndC, 66
```

```
     '// Start previewing the image from the camera

     capPreview lwndC, True

     '// Resize the capture window to show the whole image

     ResizeCaptureWindow lwndC

'         Bsent = True

'         'llamada a la funcion que nos devolvera el frame.

'         capSetCallbackOnFrame lwndC, AddressOf MyFrameCallback

'

   End If

  SetWindowPos Me.hWnd, HWND_TOPMOST, 0, 0, 0, 0, SWP_NOSIZE Or SWP_NOMOVE

End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer

 If EnCallback = False Then capSetCallbackOnFrame lwndC, vbNull

End Sub

Private Sub Form_Unload(Cancel As Integer)

     '// Disable all callbacks

     capSetCallbackOnError lwndC, vbNull

     capSetCallbackOnStatus lwndC, vbNull

     capSetCallbackOnYield lwndC, vbNull

     If EnCallback = False Then capSetCallbackOnFrame lwndC, vbNull
  capSetCallbackOnVideoStream lwndC, vbNull
      capSetCallbackOnWaveStream lwndC, vbNull
      capSetCallbackOnCapControl lwndC, vbNull
End Sub
Private Sub mnuDisplay_Click()
'  /*
'   * Display the Video Display dialog when "Display" is selected from
'   * the menu bar.
'   */
    capDlgVideoDisplay lwndC
End Sub
Private Sub mnuFormat_Click()
' /*
'   * Display the Video Format dialog when "Format" is selected from the
'   * menu bar.
'   */
    capDlgVideoFormat lwndC
    ResizeCaptureWindow lwndC
End Sub
Private Sub Salir_Click()
```

```
Cerrando = True
      Do
      DoEvents
   Loop Until EnCallback = False
    Call Form_Unload(0): Unload Me
   End Sub
```

APPLICATIONS OF OUR PROJECT:

In the hazardous environment the applications of robot in inspection is a growing area.Quality control in industrial manufacturing operation is essential in the competitive world. A manual inspection of the finished product as well as the in-process parts is a tedious and monotonous task for a human operator. In large quantity production,100% inspection is impossible. This PC control robot plays a significant role in making 100% inspection possible in the hazardous environment.

Our project spy robot can be used for inspection in

- pharmaceutical industry
- textile industry
- chemical industry
- mining industry
- construction industry
- energy sector

CONCLUSION AND FUTURE EXPANSION:

CONCLUSION:

We have designed and fabricated "SPY ROBOT"– A robot for wireless video and audio transmission" which could replace human beings in hazardous and unsafe environment .By adding several others features, this robot could be made more efficient.

FUTURE EXPANSION:

In future several features can be added to our project to make it more efficient and reliable.

- Making our Telerobot into Autonomous Robot
- Attach camera which can cover 360 degrees
- Adding few more sensors like Obstacle sensor, IR sensor

Provide Dual way Communication

Of the robots in the world today about 90% are found in industries. These robots are referred to as industrial robots and are regarded as "Steel collar workers". In this PC controlled the robot the additional features such as

- handling dangerous objects
- space exploration
- under-sea exploration
- nuclear research

- geological exploration
- play football,cricket
- traffic control
- fire fighting
- assembling products
- polishing and cutting
- tasks involving danger to humans
- material handling
- material transfer applications
- .machine loading and unloading application
- processing applications
- arc welding

APPENDIX:

INSTALLATION PROCEDURE:

Step 1: Install the TV tuner card using driver CD

Step 2: Connect the yellow pin of RF receiver to the TV tuner card

Step 3: Connect the black of RF receiver to +12V power supply

Step 4: Connect the camera to +9V power supply

Step 5: Tune the RF receiver to get clear audio and video.

REFERENCES

1. Ajay V. Deshmuk “Microcontroller Programming and Interfacing”, Tata McGraw Hill, New Delhi.
2. www.rentron.com
3. www.atmel.com
4. http://www.robotics.com/robots.html
5. www.logix4u.net
6. www.kmitl.ac.th/~kswichit/ISP-Pgm3v0/ISP-Pgm3v0.html
7. http://www.8052.com/users/AT89S51InSystemProg/
8. http://www.esacademy.com/automation/docs/c51primer/c51prim.htm
9. http://www.faculty.ucr.edu/~currie/roboadam.htm
10. www.zen22142.zen.co.uk/Circuits/rf/tv.htm