ROBOT WRANGLER

A Robot for Wireless Video Transmission

A Project Report

Submitted By

T.Mathesh	-	71203105026
R.Prabahar	-	71203105034
Y.Srinivasan	-	71203105046

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

ANNA UNIVERSITY :: CHENNAI 600 025

APRIL – 2007

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report entitled **"ROBOT WRANGLER-A Robot for Wireless Video Transmission"** is a bonafide work of

T.Mathesh	-	Register No	71203105026
R.Prabahar	-	Register No	71203105034
Y.Srinivasan	-	Register No	71203105046

Who carried out the project under my supervision.

KSubramanian

Signature of the DEAN/HOD

(Prof.K. Ragupathy Subramanian B.E(Hons),M.Sc.)

DEAN/ HOD

Signature of the guide

(Mr. C. Udhaya Shankar, M.Tech)

Lecturer

KSubramanian

Internal Examiner 25/4/07

V Rajkumar 25/4/07

External Examiner

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006

ABSTRACT

The project "**Robot Wrangler**" deals with design and fabrication of a robot to transmit the Video signals captured by the camera placed on the robot to a PC located far away using wire less transmission

Our project has two main sections Control Section and Robot Section. The computer placed at remote place sends the signals through RF transmitter in the control section. The RF receiver placed on the robot receives the control signals and feeds the signal to the Micro controller. The microcontroller which is Pre-programmed receives the input from the control section and activates the driver circuit accordingly to control DC motor which in turn controls the movement of the robot. There's also a flame sensor in the robot which senses flame in its path and trigger a buzzer circuit.

The video signal captured by the camera placed on robot is transmitted to PC using RF transmission technique. The transmitted signal is then received using TV tuner card and monitored through visual basic front end.

The system is designed for use in dangerous and hazardous environment such as nuclear & chemical plants, near oil wells and in military operations where human entry is limited. It can also be used in space vehicle to capture the images in other planets and in Milky Way provided transmitter and receiver section used should be of superior range.

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 allowing us 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

TITLE	PAGE NO
Bonafide Certificate	ii
Abstract	iii
Acknowledgement	iv
List of Tables	vii
List of Figures/Photos	viii

CHAPTER 1 INTRODUCTION

1.1 Need for the project.	01
1.2 Objective of the project.	01
1.3 Features of the project.	01
1.4 Scope of the project.	02
1.5 Block diagram.	02
1.6 Project overview.	02

CHAPTER 2 PC INTERFACE AND CONTROL UNIT FOR ROBOT

2.1 Block diagram.	08
2.2 Power Supply Unit.	10
2.3 VB Front end.	12
2.4 Parallel Port interface.	14
2.5 Encoder and Transmitter	19

CHAPTER 3 VIDEO TRANSMISSION ROBOT

3.1 Block diagram 23

3.2 Receiver and decoder 24

3.3 Micro controller-AT89c51 27

3.4 Microcontroller Programming 33

3.5 Video Transmitter Section 42

CHAPTER 4 VIDEO RECEPTION IN PC

4.1. Hardware unit 45

4.2. Interfacing TV tuner card with PC 46

4.3. VB Coding for video reception 49

CONCLUSION & FUTURE ENHANCEMENTS 58

APPENDICES 59

REFERENCES 72

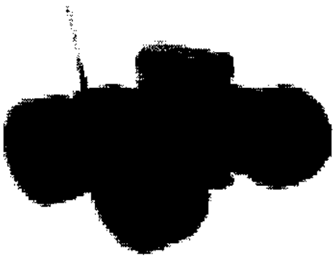
LIST OF TABLES

Table	Title	Page No.
2.1	Operations based on HEX code	17
3.1	Robot movement based on motor direction	29

LIST OF FIGURES

FIGURE NO	TITLE	PAGE.NO
1.1	Overall Block Diagram	7
2.1	Control Section Hardware Unit	9
2.2	Power Supply Unit	10
2.3	Power Supply Circuit	11
2.4	VB Front End	12
2.5	Parallel Port Layout	15
2.6	Printer Port Properties	16
2.7	Parallel Port Interfacing with Encoder	17
2.8	Encoder Pin Details	19
2.9	Transmitter Pin configuration	20
2.10	Encoder with RF Transmitter	21
3.1	Video Transmission hardware unit	23
3.2	Decoder Pin Details	24
3.3	RF Receiver Pin Details	25
3.4	Decoder with RF Receiver	26
3.5	Dc Motor & Relay Circuit	30
3.6	Micro-controller with DC motor and Relay Unit	31
3.7	Fire Sensor Circuit	32
3.8	ATMEL 89c51 Pin Details	33
3.9	Flash 89 Programmer Screen	36
3.10	Flowchart For DC motor Control	38
3.11	Video Transmitter Circuit	43
4.1	Hardware Ůnits - Video Reception	45
4.2	TV Tuner Card Layout	46

1. INTRODUCTION



1.1 Need for the project.

1.2 Objective of the project.

1.3 Features of the project.

1.4 Scope of the project.

1.5 Block diagram.

1.6 Project overview.

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 OBJECTIVE OF OUR PROJECT:

The main objective of our project is to transmit the image captured by the robot in the survey field to system and also to control the entire operation of the robot including fire detection.

1.3 FEATURES OF THE PROJECT:

In the project "Robot Wrangler – Robot for wireless video transmission" we have designed and developed a robot, which could be used 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.

The operator stays in a secure atmosphere and guides the robot in hazardous environment by using a PC. For controlling the robot control signals are transmitted from PC – parallel port to robot section via RF frequency. The data to the parallel port is controlled by using Visual Basic as front end. Each buttons used for direction control (forward, reverse, left, right & stop) has a different set of data related with it. Once the button is clicked the corresponding data is transmitted to robot.

Similarly, video captured by the robot is transmitted in air in the form of video signal using a video transmitter circuit. By using an antenna we could capture the video signal and give it to PC via TV Tuner Card. By tuning for the required frequency in the tuner card we could see the video that is transmitted.

There's also a provision in the VB screen, which eliminates the need for having separate screen to view the transmitted video. The coding is written such that the source for the video is form the tuner card input. So the operator could see the survey field in the screen and guide the robot accordingly.

The parallel port interfacing, the signal transmission and reception concepts, video transmission circuit details are dealt in the later chapters.

1.4 SCOPE 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 Robot Wrangler 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.

1.6 PROJECT OVERVIEW

Robot Wrangler 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.

Let us have a brief look at them.

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 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 434 MHz.

VIDEO 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.

These are the components in the control section.

ROBOT SECTION :

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

- Signal reception section,
- Microcontroller (89C51),
- Driver circuit,
- Fire sensor circuit,
- Camera and Video transmitter section.

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 fire sensor.

Advantages of using microcontroller :

- ✓ In built storage,
- ✓ PCB size reduced,
- ✓ Low design cost

This module receives the input from the receiver section and processes the data.It controls the movement of robot. It has many features like On - Chip Flash Program Memory, On - Chip Data RAM, Watch dog timer etc. These are dealt in the later sections.

)RIVER CIRCUIT:

The driver circuit is used to control the two DC motors in the Robot thereby controlling :he 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

FIRE SENSOR CIRCUIT :

The fire sensor in the Robot section senses fire in its path. If there is any fire, the sensor senses it, halts the robot and gives a continuous alarm (Buzzer) till the fire ceases. The sensor circuit also triggers the relay. The required operation such as fire quenching can be implemented as per needs in the relay circuit in future.

VIDEO TRANSMITTER SECTION :

The video transmission circuit is used to transmit the video captured by the camera, in the survey field to the PC at remote place. 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.

These are the modules in the robot section.

1.5 BLOCK DIAGRAM OF THE PROJECT

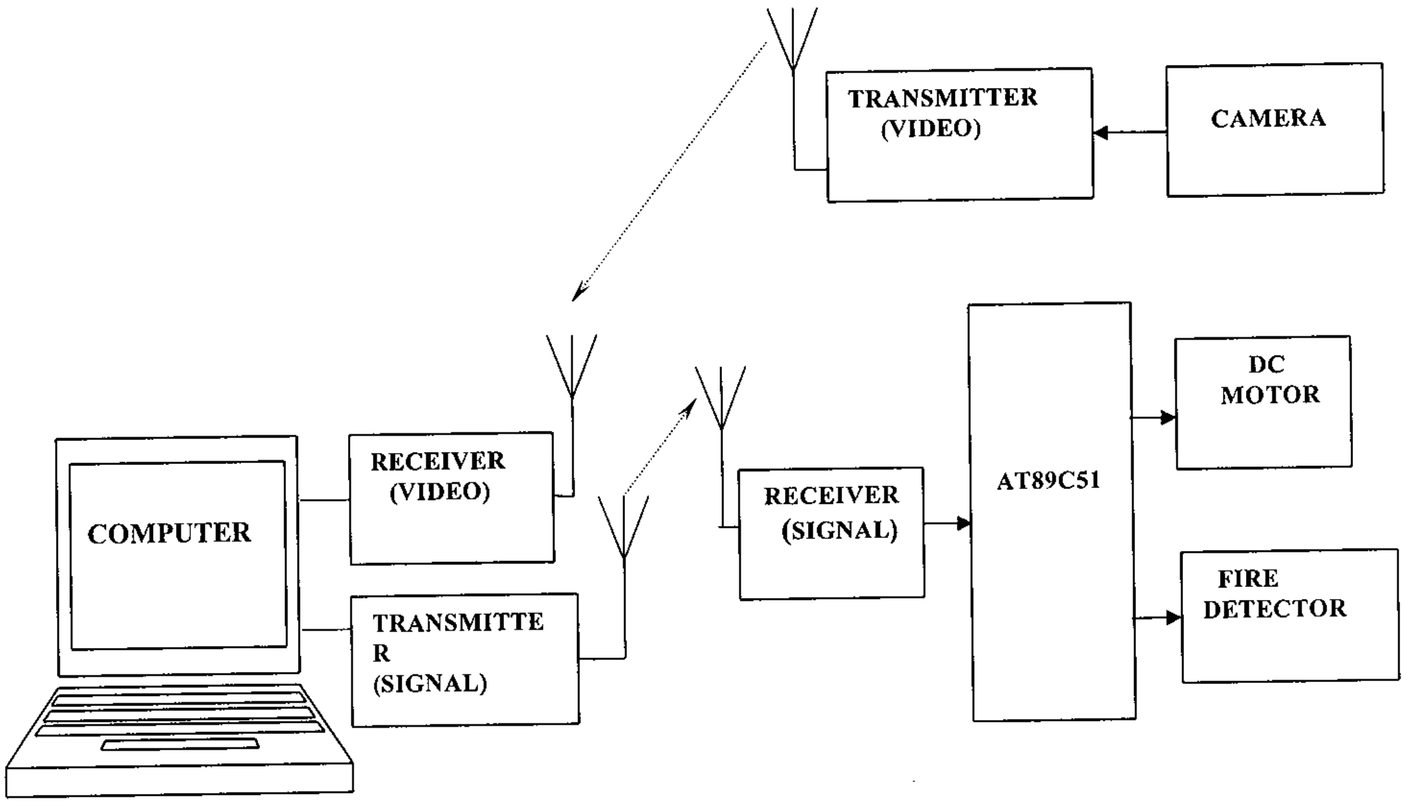
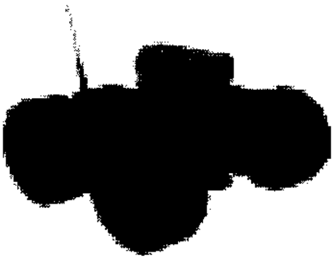


Fig 1.1 Overall Block Diagram

2. PC INTERFACE AND CONTROL UNIT FOR ROBOT



2.1 Block diagram.

2.2 Power Supply Unit.

2.3 VB Front end.

2.4 Parallel Port interface.

2.5 Encoder and Transmitter.

2.1 SYSTEM DESCRIPTION OF CONTROL SECTION

2.1.1 HARDWARE UNIT:

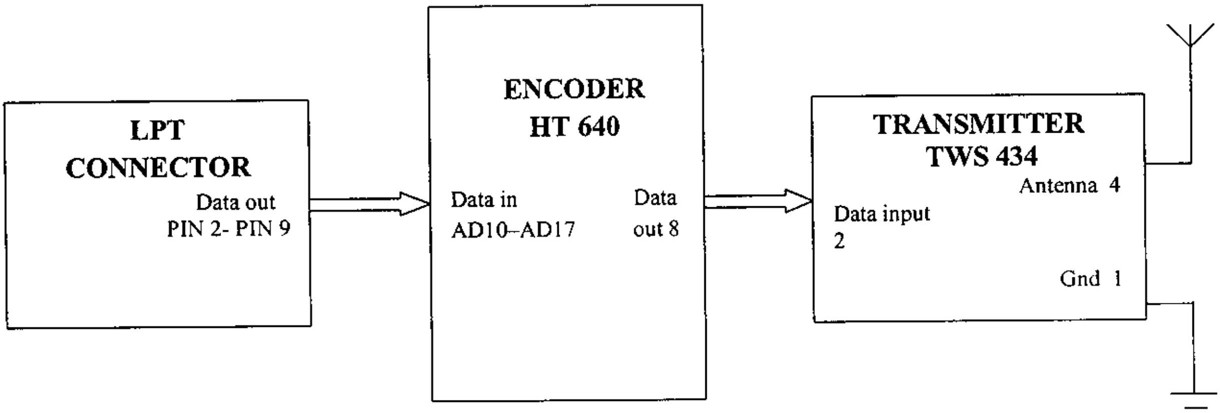


Fig 2.1 Control Section – Hardware Unit

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 2.2. 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.

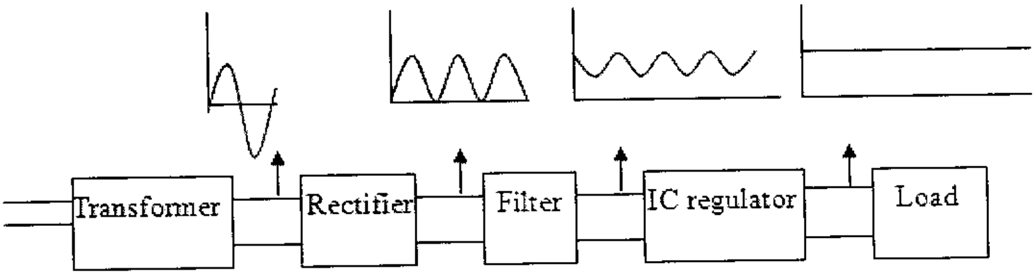
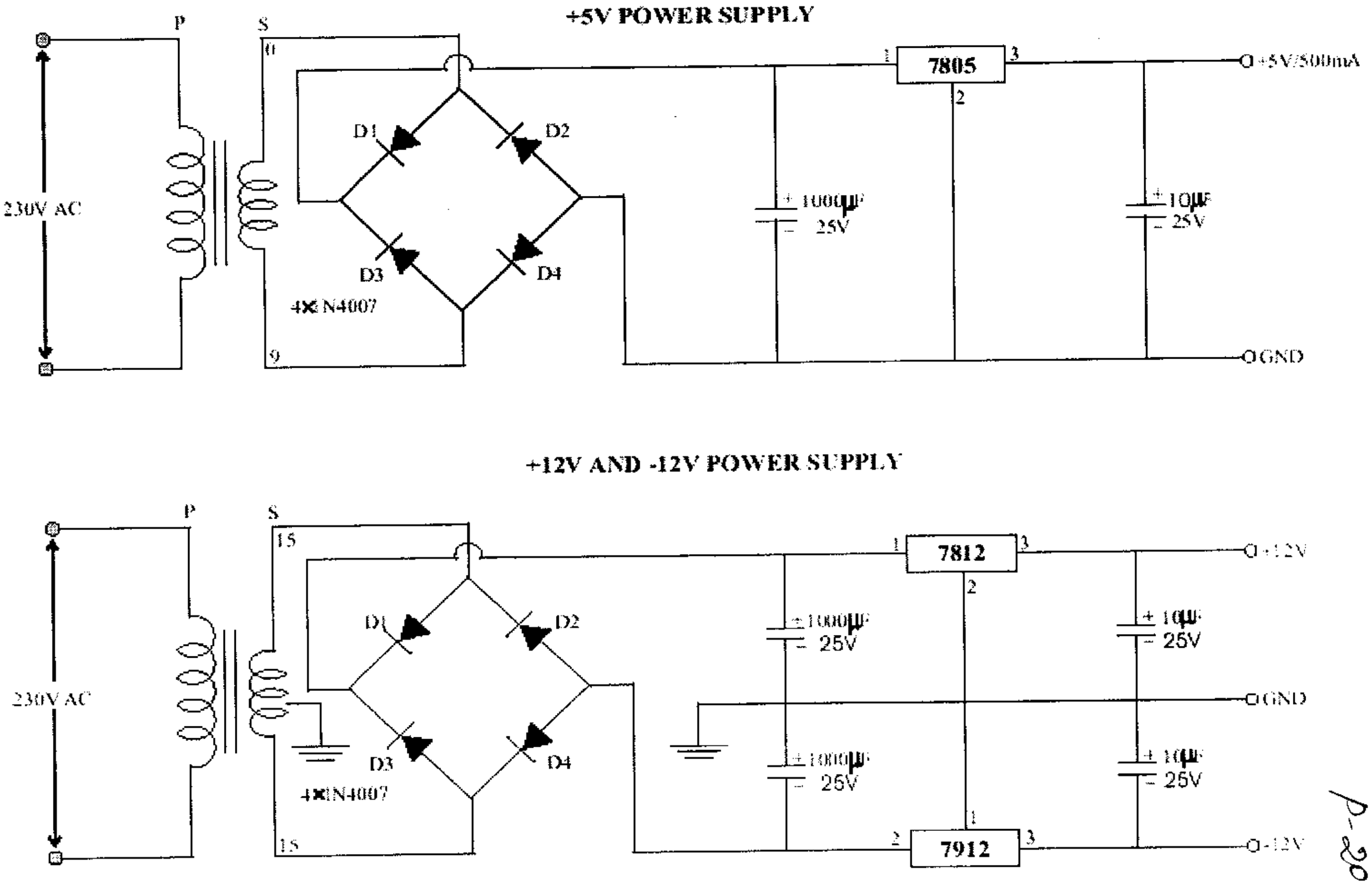


Fig 2.2 Power Supply Unit

2.2.2 CIRCUIT DIAGRAM OF POWER SUPPLY UNIT:



+5V POWER SUPPLY

+12V AND -12V POWER SUPPLY

Fig 2.3 Power Supply Circuit

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

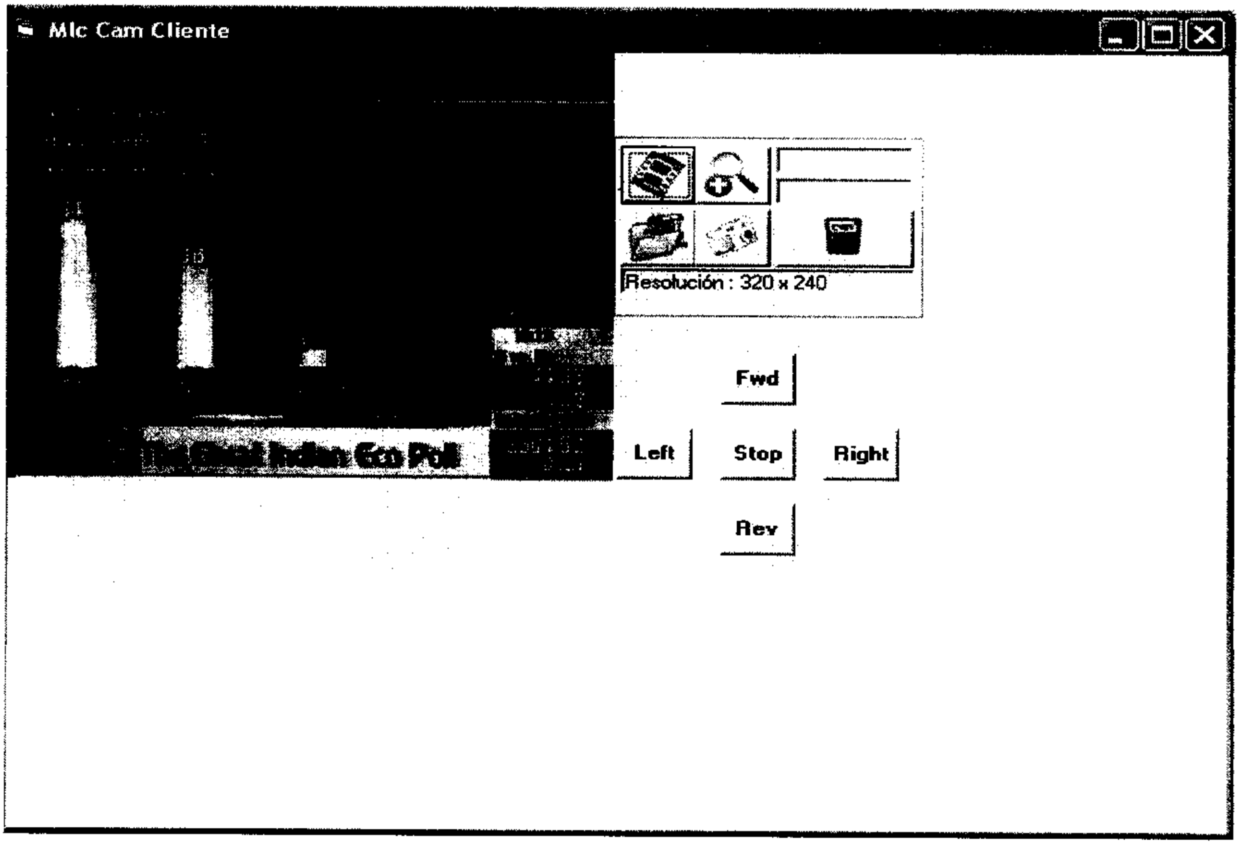


Fig 2.4 VB Front End

2.3.3 VB CODING :

```
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

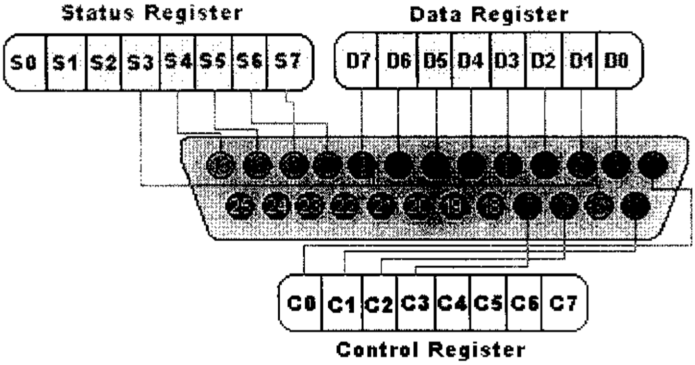


Fig 2.5 Parallel Port Layout

Status lines -exchange of messages, state indicators from the printer to the PC

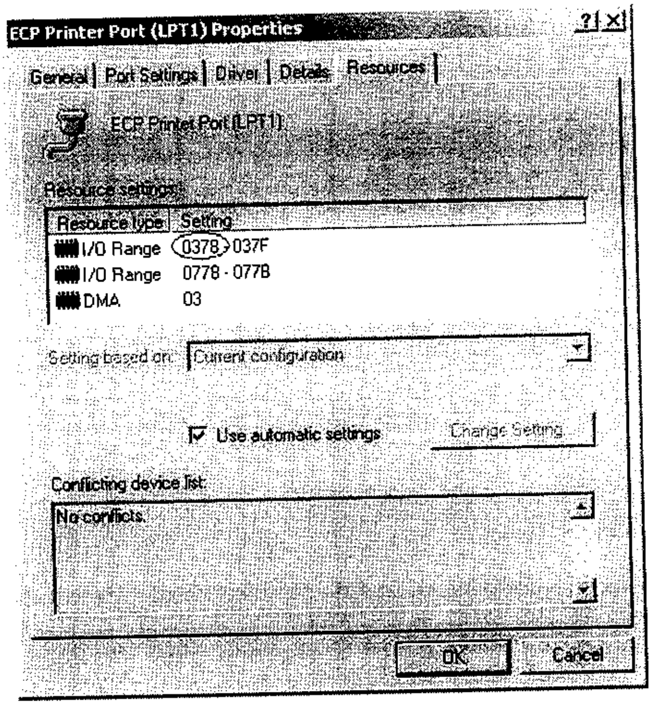
lines of data -gives the data of impression of the PC toward the printer and only in that address. Each one of these lines (control, state, data) it can be indexed in an independent way by means of a registration.

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.



ECP Printer Port (LPT1)

Resource settings:

Resource type	Setting
I/O Range	0378 - 037F
I/O Range	0778 - 077B
DMA	03

Setting based on: Current configuration

☑ Use automatic settings

Conflicting device list:

No conflicts.

Fig 2.6 Printer Port Properties

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

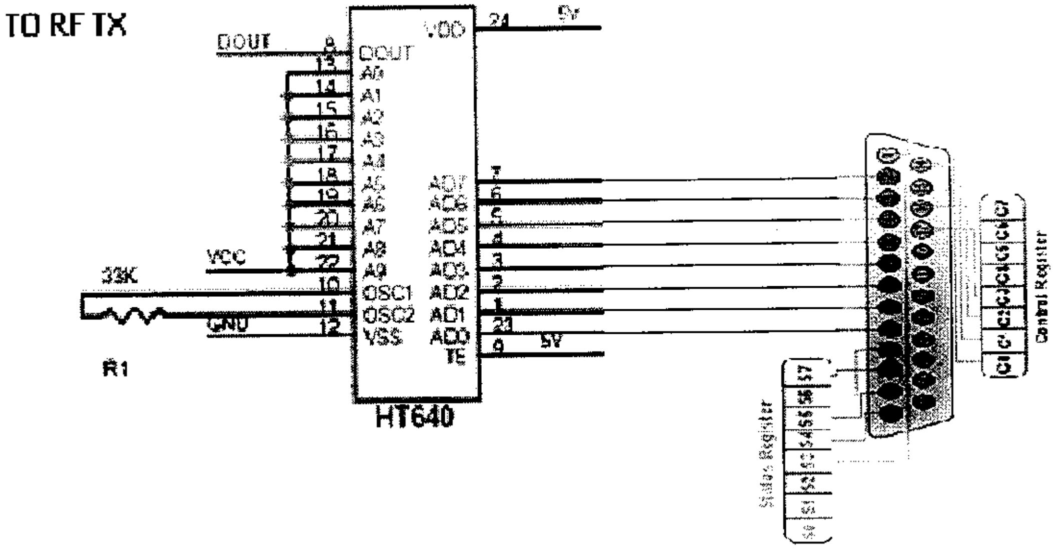


Fig 2.7 Parallel Port Interfacing with Encoder

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

Table 2.1 - Operations based on HEX code

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

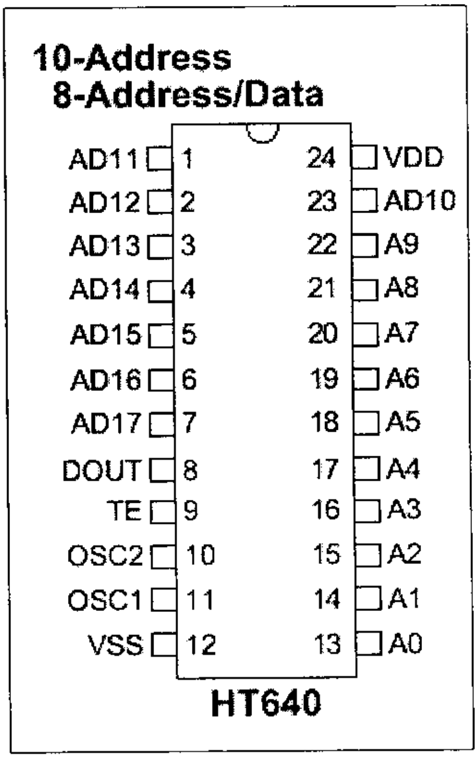


Fig 2.8 Encoder Pin Details

2.5.3 RF TRANSMITTER:

Circuit Diagram:

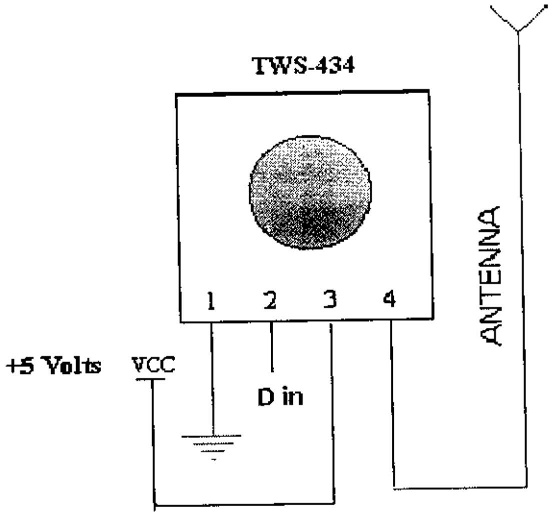


Fig 2.9 Transmitter Pin configuration

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

- It is driven by a single 9V supply from a battery
- It has an 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:

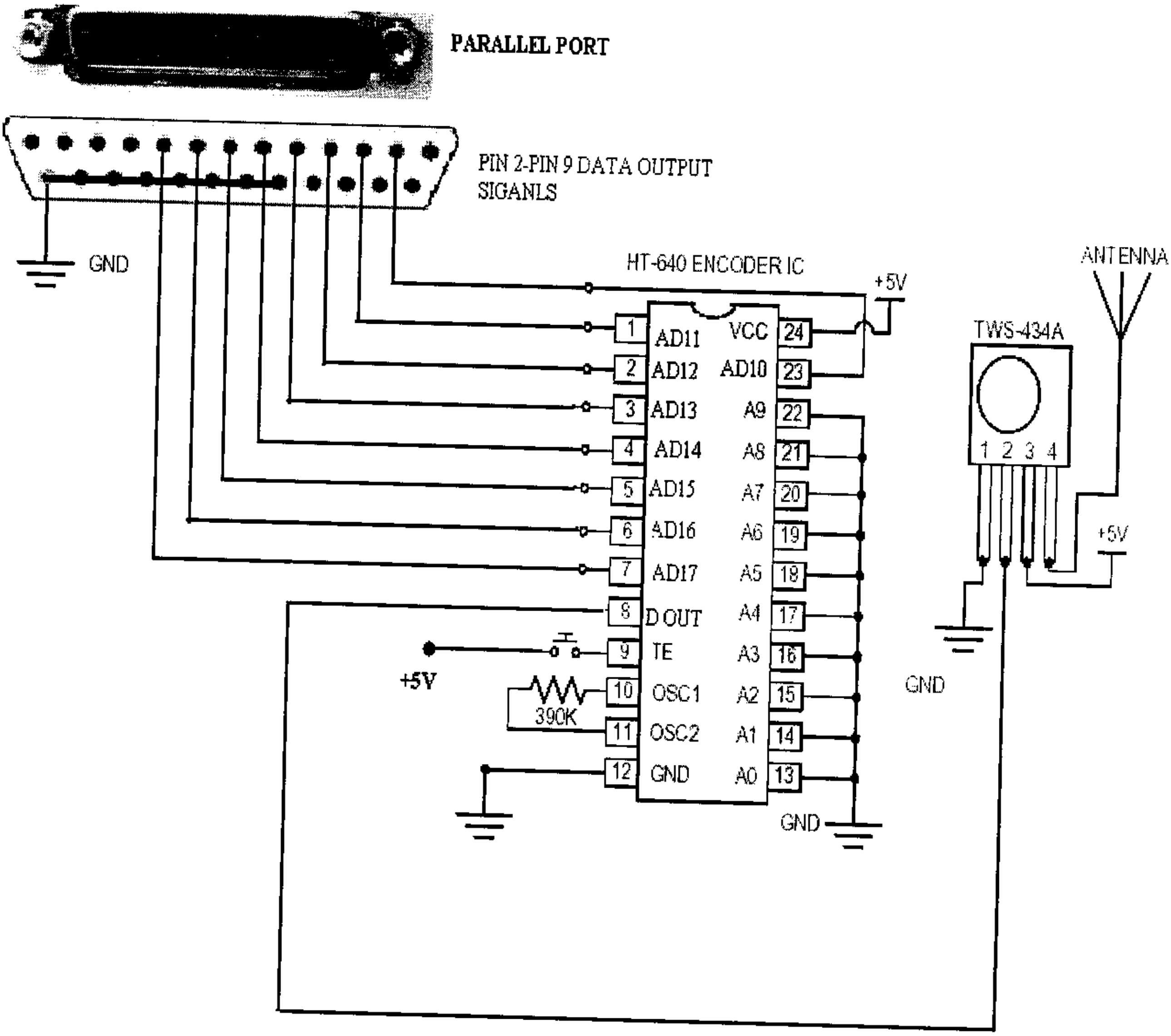
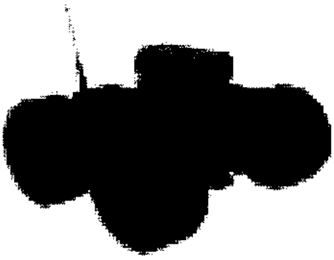


Fig 2.10 Encoder with RF Transmitter

3. VIDEO TRANSMISSION ROBOT



3.1 Block diagram

3.2 Receiver and decoder

3.3 Micro controller-AT89c51

3.4 Microcontroller Programming

3.5 Video Transmitter Section

3. HARDWARE UNIT

YSTEM DESCRIPTION:

ROBOT SECTION:

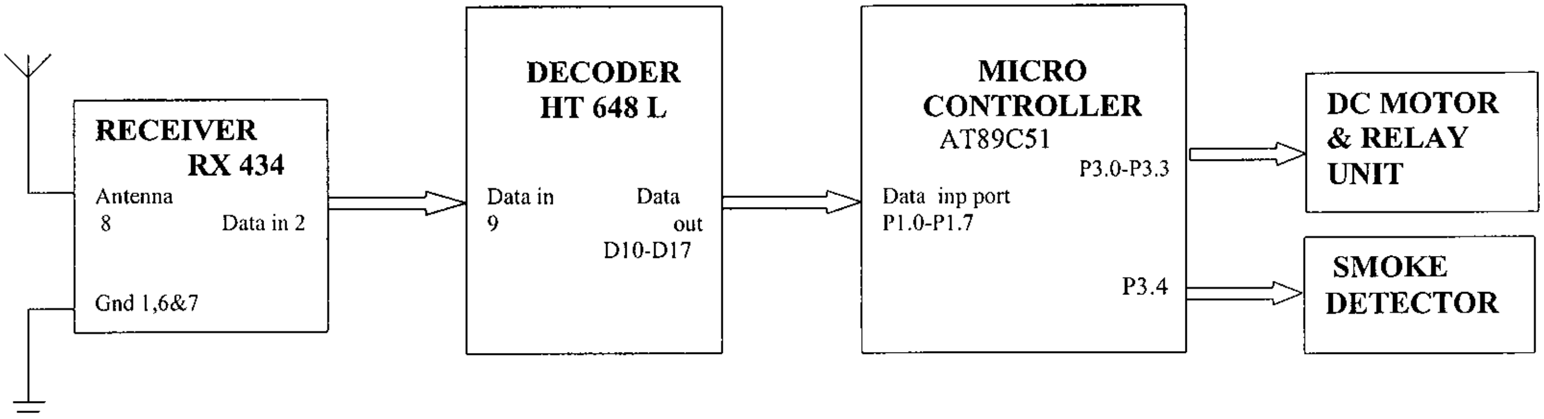


Fig 3.1 Video Transmission hardware unit

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

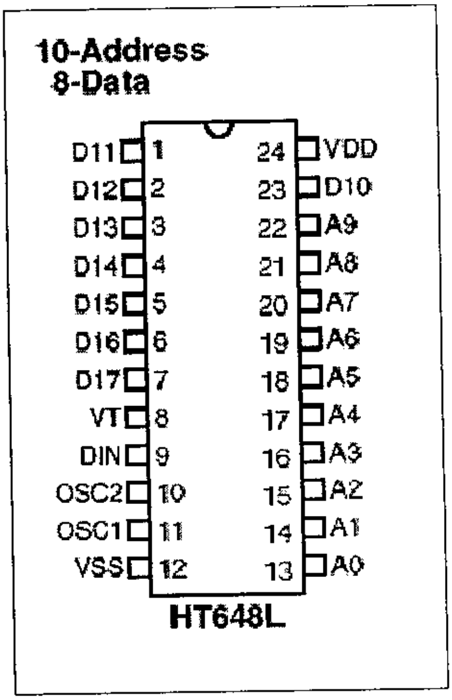
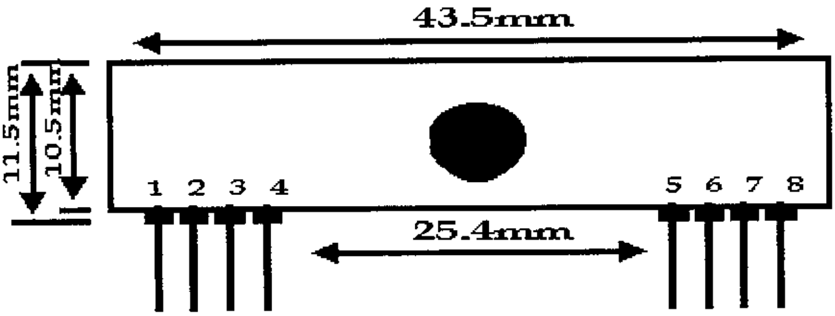


Fig 3.2 Decoder Pin Details

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 Pin Details

The RF receiver works under the concept of ASK modulation.

- It is driven by a single 9V supply from a battery
- It has 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

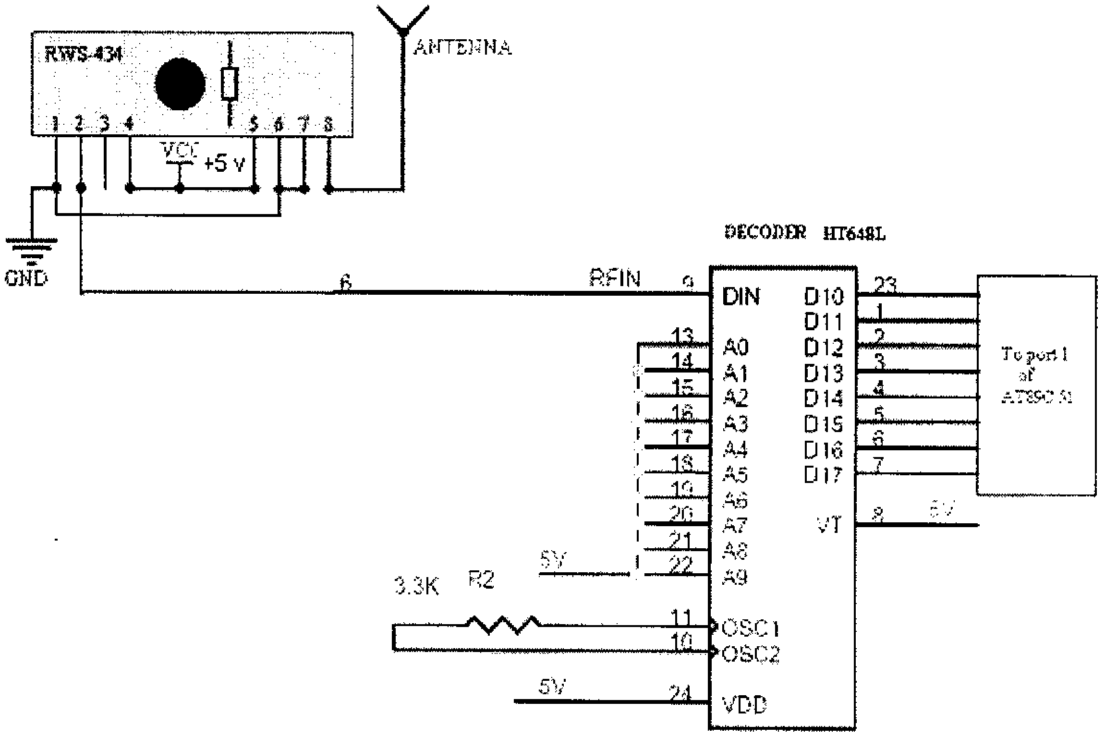


Fig 3.4 Decoder with RF Receiver

3.3 MICRO CONTROLLER SECTION

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

There are four basic aspects of a microcontroller: speed, size, memory, and other. Speed s designated in clock cycles, and is usually measured in millions of cycles per second (Megahertz, MHz). The use of the cycles varies in different MCUs, affecting the usable speed of he processor. MCUs come in 4-, 8-, 16-, and 32-bits, with 8-bit MCUs being the most common size. MCUs count most of their ROM in thousands of bytes (KB) and RAM in single bytes. Many MCUs use the Harvard architecture, in which the program is kept in one section of memory (usually the internal or external SRAM). This in turn allows the processor to access the separate memories more efficiently.

The fourth aspect of microcontrollers, referred to as "other", includes features such as a dedicated input device that often (but not always) has a small LED or LCD display for output. A microcontroller also takes input from the device and controls it by sending signals to different components in the device. Also the program counter keeps track of which command is to be executed by the microcontroller.

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. Fire Alarm 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.

Table 3.1- Robot movement based on motor 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:

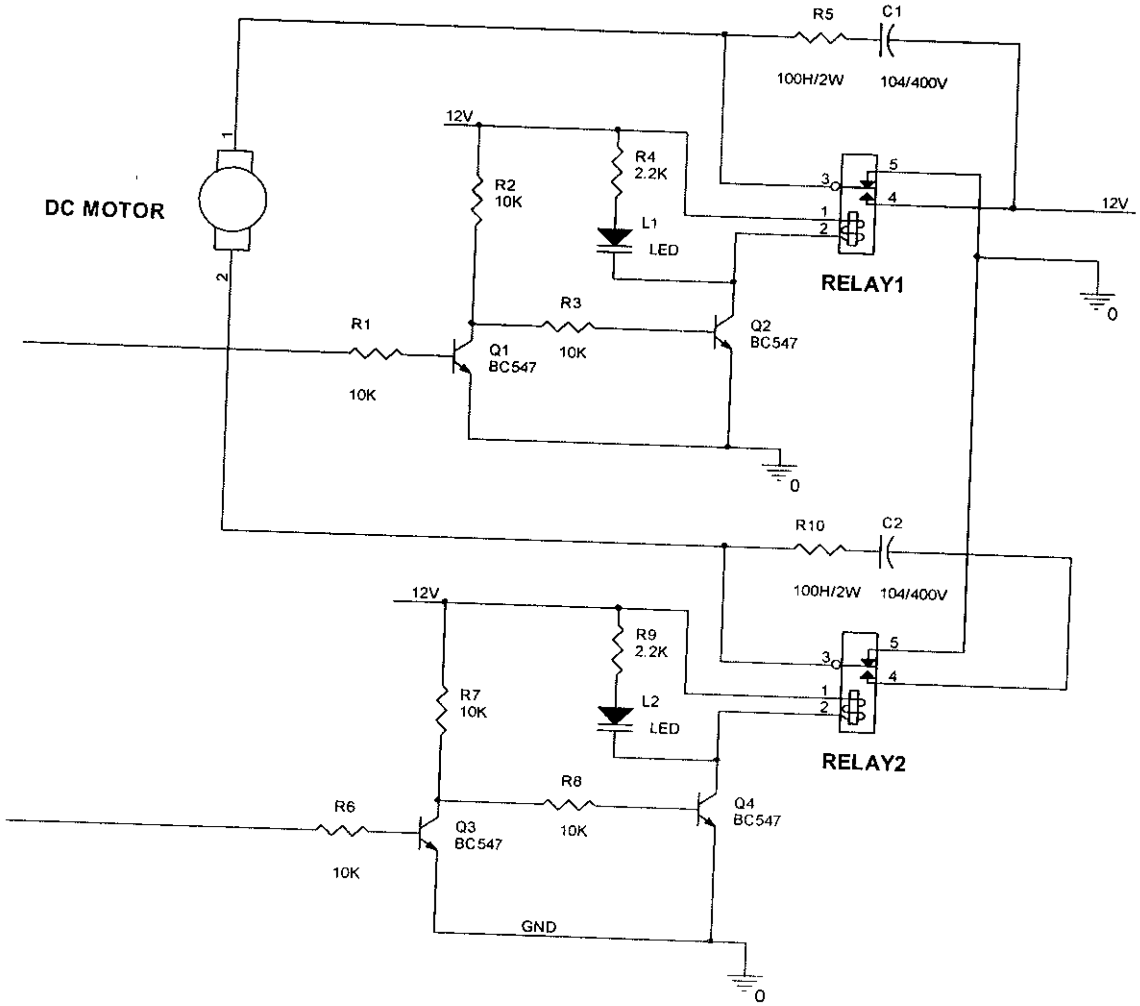


Fig 3.5 Dc Motor & Relay Circuit

3.3.2.3 Micro-controller with DC motor and Relay Unit

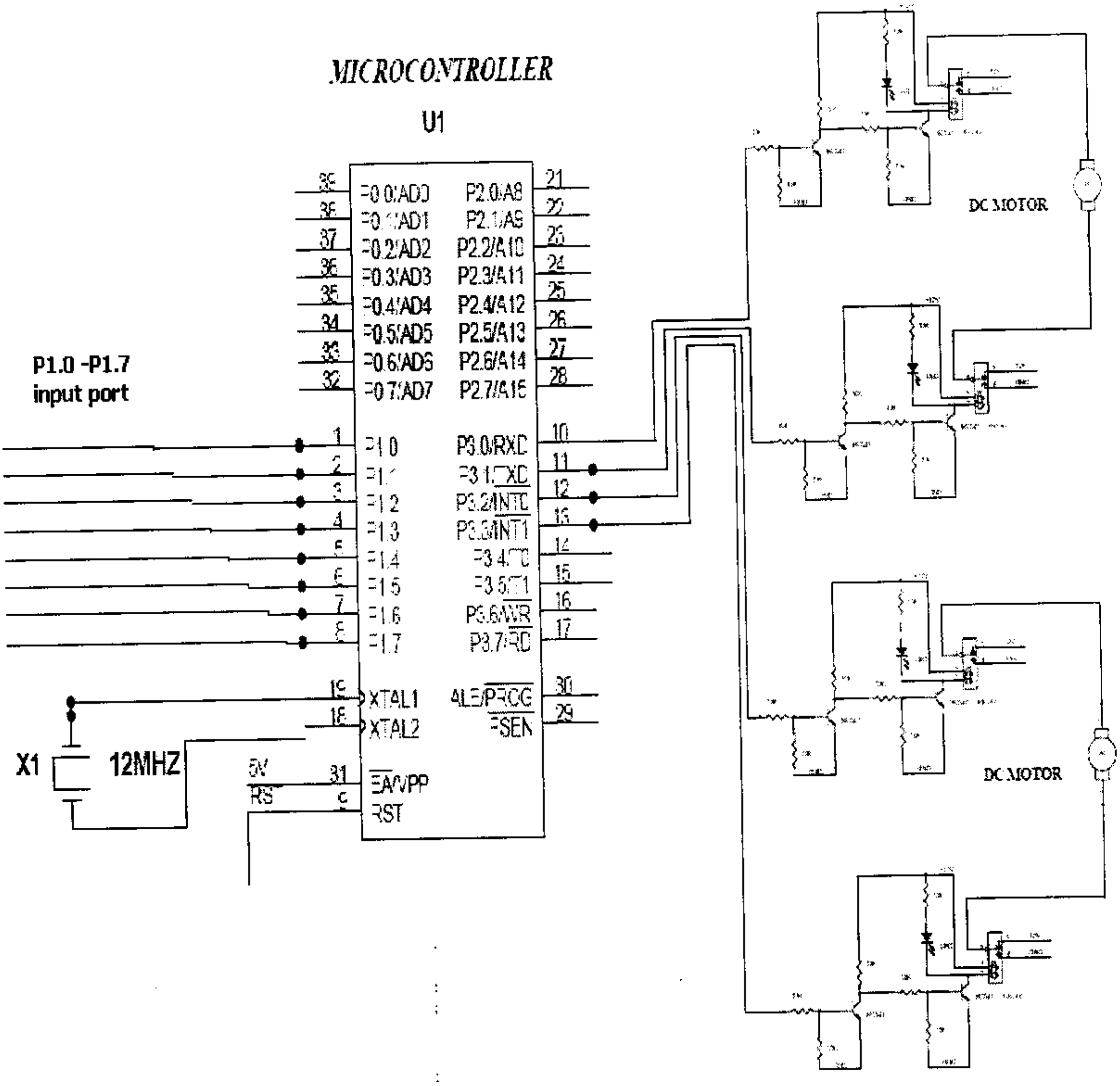


Fig 3.6 Micro-controller with DC motor and Relay Unit

3.3.2.4 FIRE SENSOR CIRCUIT

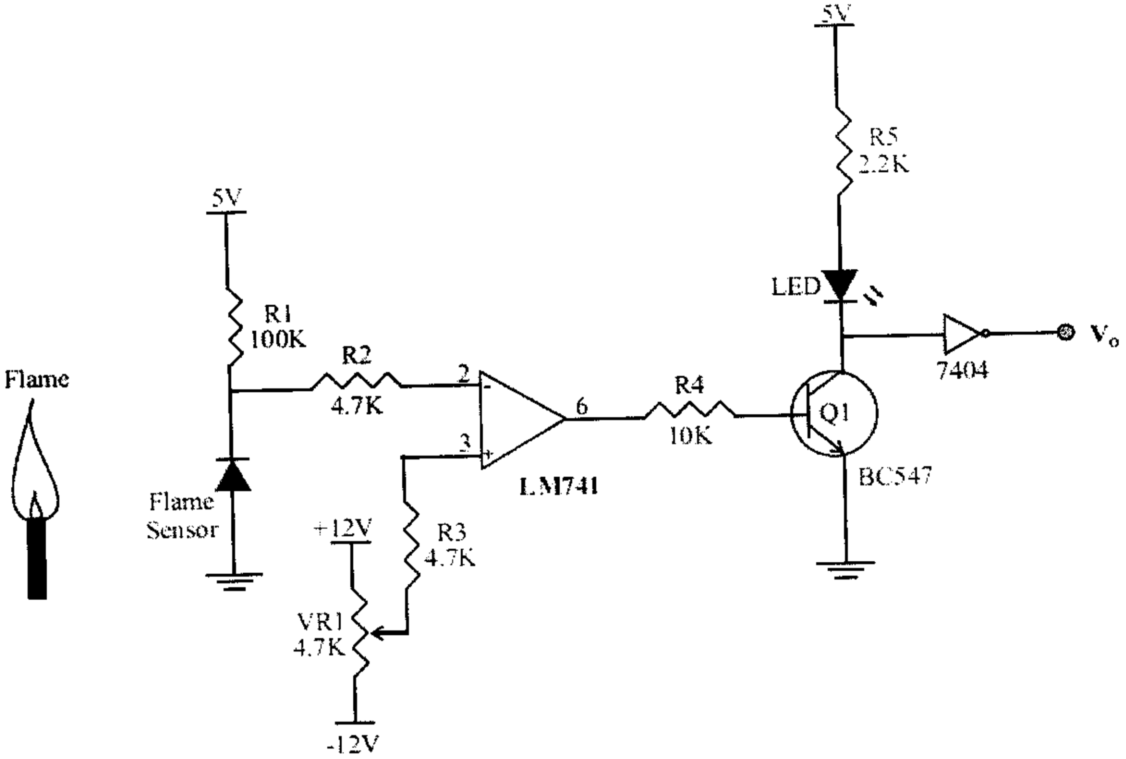


Fig 3.7 Fire Sensor Circuit

When there is a flame in the survey field the telerobot senses it through the flame sensor and the output of differentiator (LM 741) becomes high. Now the transistor starts conducting and so the input to the NOT gate goes low. The output from NOT gate is then given to the microcontroller (ATMEL 89C51) pin 16. The microcontroller processes the input and then triggers the alarm circuit.

3.3.2.5 ATMEL 89C51 Pin configuration:

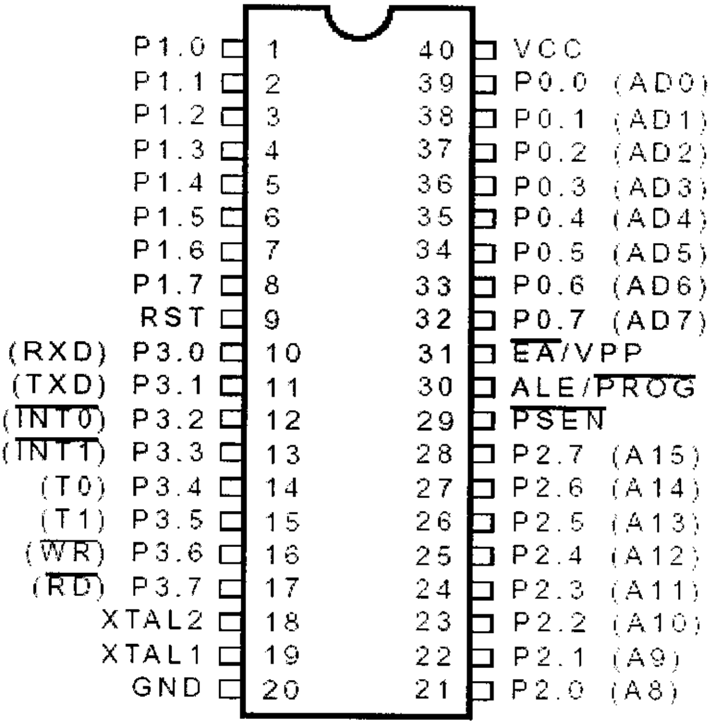


Fig 3.8 ATMEL 89c51 Pin Details

3.4 MICROCONTROLLER PROGRAMMING

3.4.1 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.

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

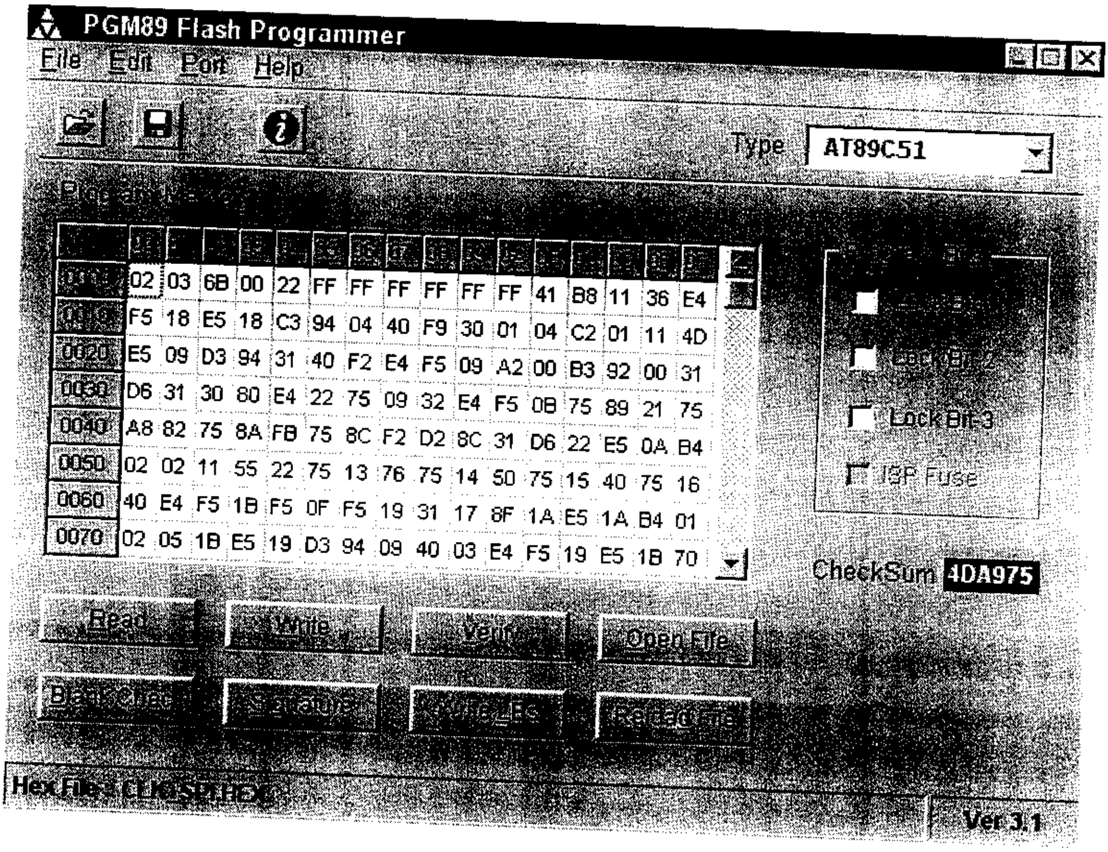
This programmer has intelligent onboard firmware and connects to the serial port. It can be used with any type of computer and requires no special hardware. The programmer is connected to the PC through serial communication port. The programmer connects to a host computer using a standard RS232 serial 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.



Type AT89C51

■ Lock Bit 1
■ Lock Bit 2
□ Lock Bit 3
□ ISP Fuse

CheckSum 4DA975

Hex File : C:\...SPI.HEX

0000	02	03	6B	00	22	FF	FF	FF	FF	FF	FF	41	B8	11	36	E4
0010	F5	18	E5	18	C3	94	04	40	F9	30	01	04	C2	01	11	4D
0020	E5	09	D3	94	31	40	F2	E4	F5	09	A2	00	B3	92	00	31
0030	D6	31	30	80	E4	22	75	09	32	E4	F5	0B	75	89	21	75
0040	A8	82	75	8A	FB	75	8C	F2	D2	8C	31	D6	22	E5	0A	B4
0050	02	02	11	55	22	75	13	76	75	14	50	75	15	40	75	16
0060	40	E4	F5	1B	F5	0F	F5	19	31	17	8F	1A	E5	1A	B4	01
0070	02	05	1B	E5	19	D3	94	09	40	03	E4	F5	19	E5	1B	70

Fig 3.9 Flash 89 Programmer Screen

3.4.2.4 BURNING PROCESS

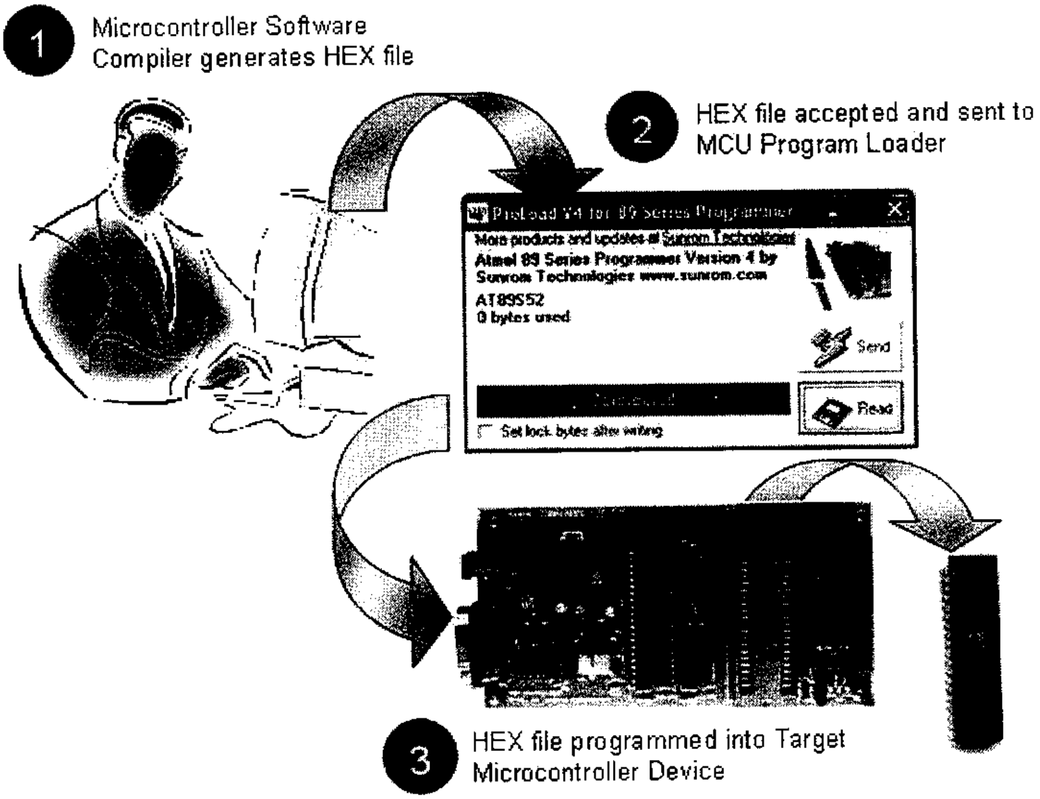
Burning is nothing but the process of loading the program into the microcontroller using suitable software. The programmer is connected to the PC through RS232 serial communication port. The process is given below

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.


1
Microcontroller Software
Compiler generates HEX file
2
HEX file accepted and sent to
MCU Program Loader
3
HEX file programmed into Target
Microcontroller Device
Send
Read

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

By this way the microcontroller can be programmed.

FLOWCHART:

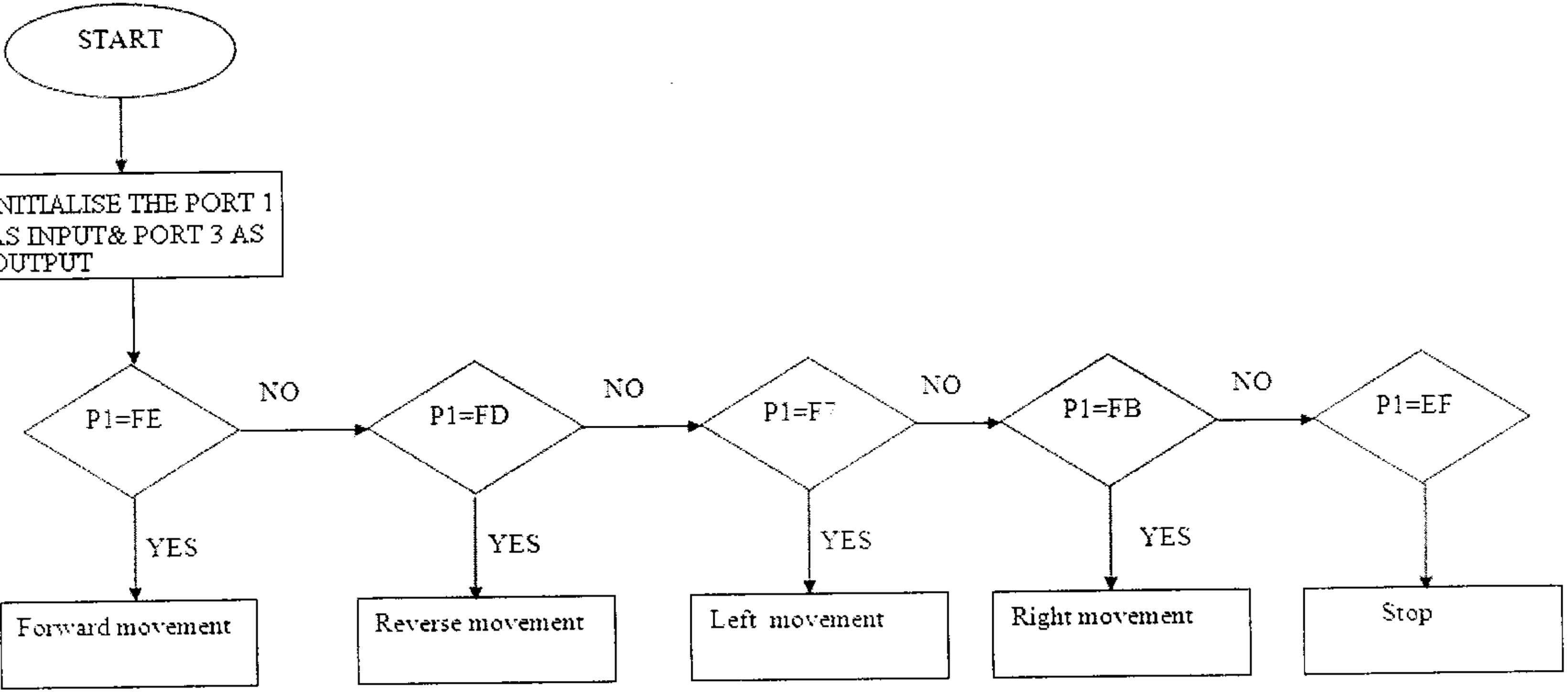


Fig 3.10 Flowchart For DC motor Control

3.4.3 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:

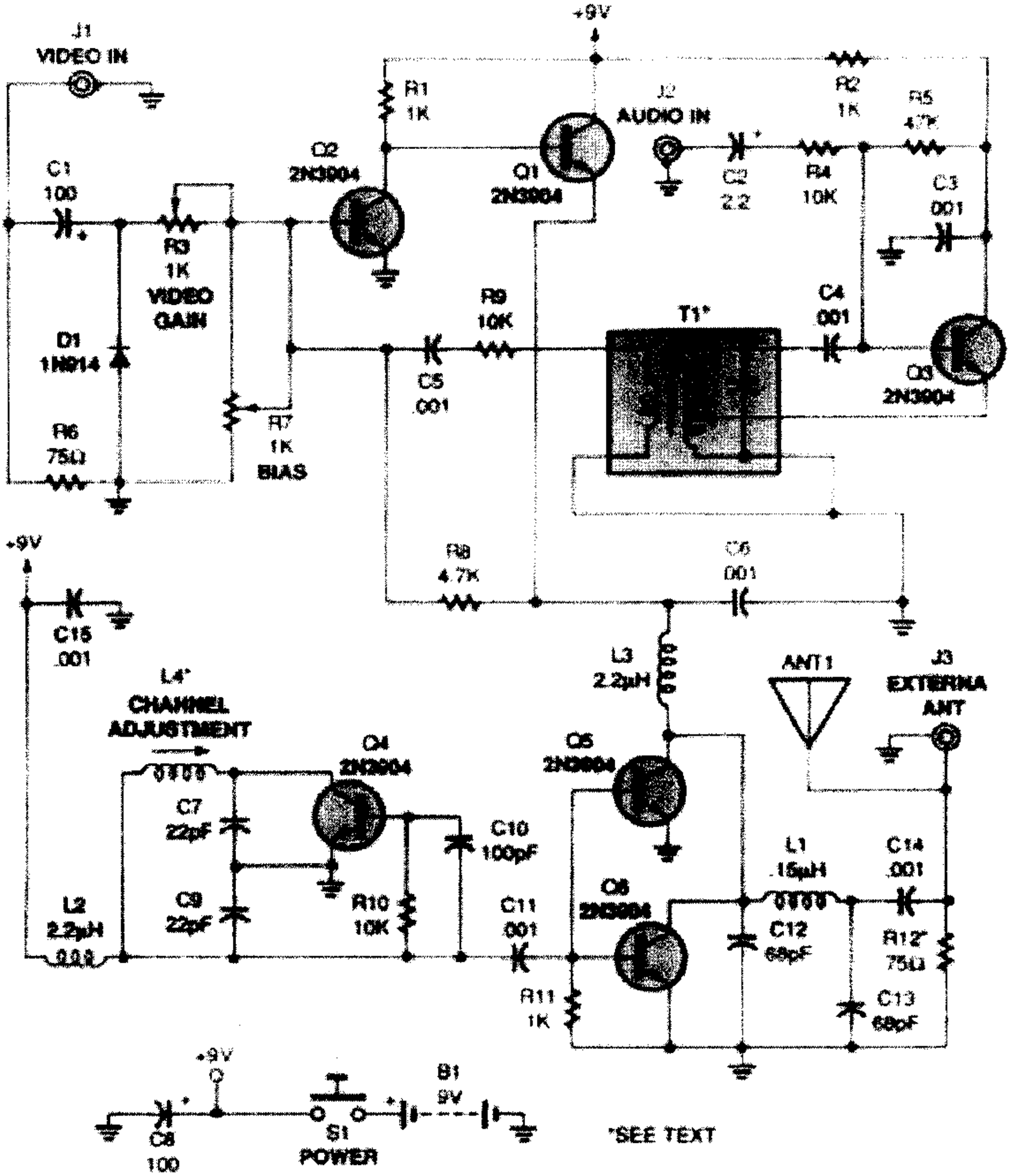
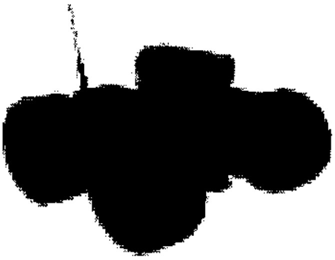


Fig 3.11 Video Transmitter Circuit

4. VIDEO RECEPTION IN PC



4.1. Hardware unit

4.2. Interfacing TV tuner card with PC

4.3. VB Coding for video reception

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.

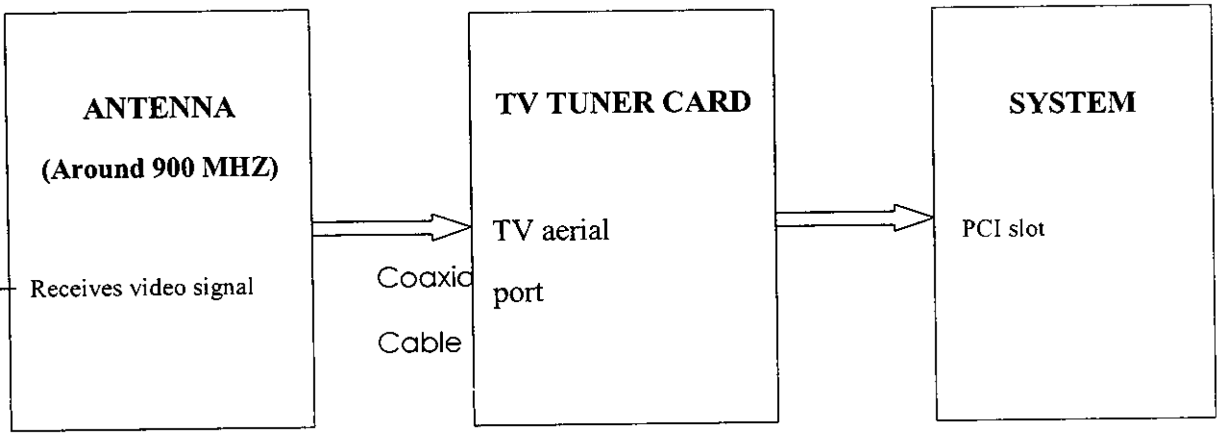


Fig 4.1 Hardware Units - Video Reception

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 interfaced to the PC via **PCI slot**. The TV tuner card is shown below.

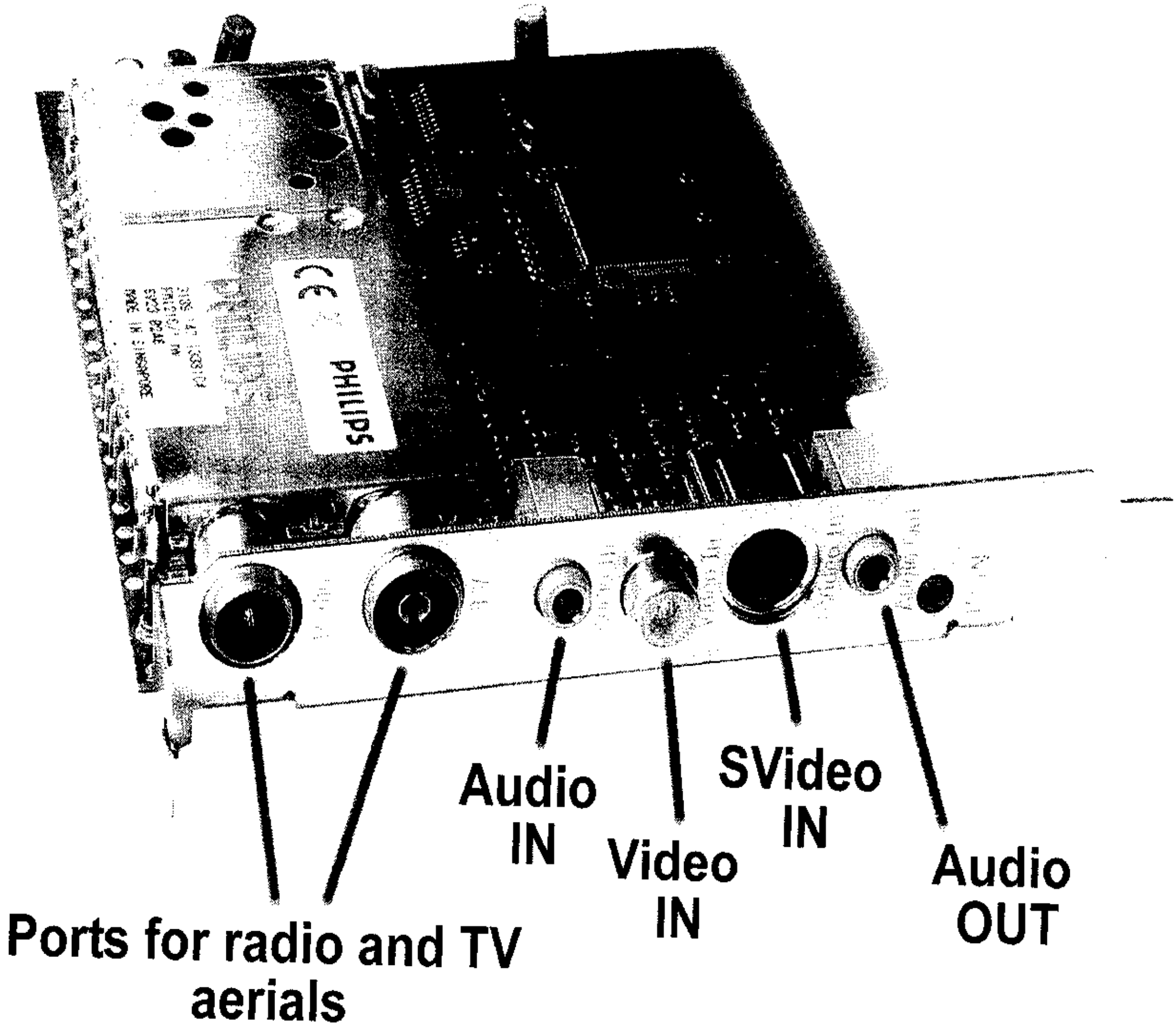


Fig 4.2 TV Tuner Card Layout

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 the 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 handle
```

```
lRes = 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
```

CONCLUSION AND FUTURE EXPANSION:

CONCLUSION:

We have designed and fabricated "ROBOT WRANGLER – A robot for wireless video 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

APPENDICES

8-Bit Microcontroller with 4K Bytes Flash

AT89C51

Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

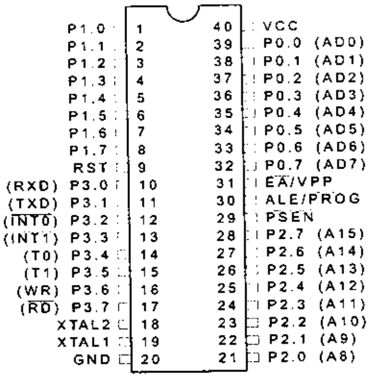
Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

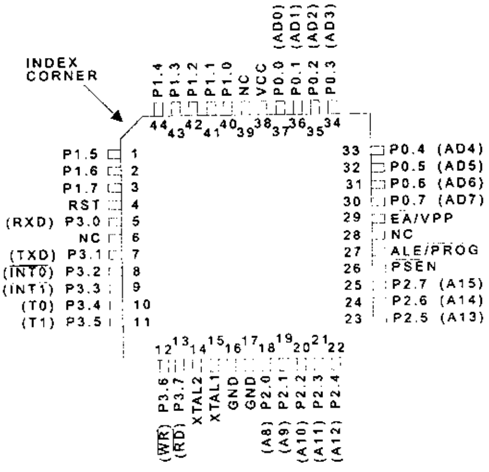
(continued)

Pin Configurations

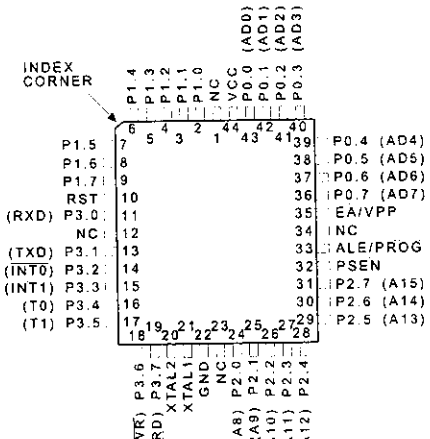
PDIP



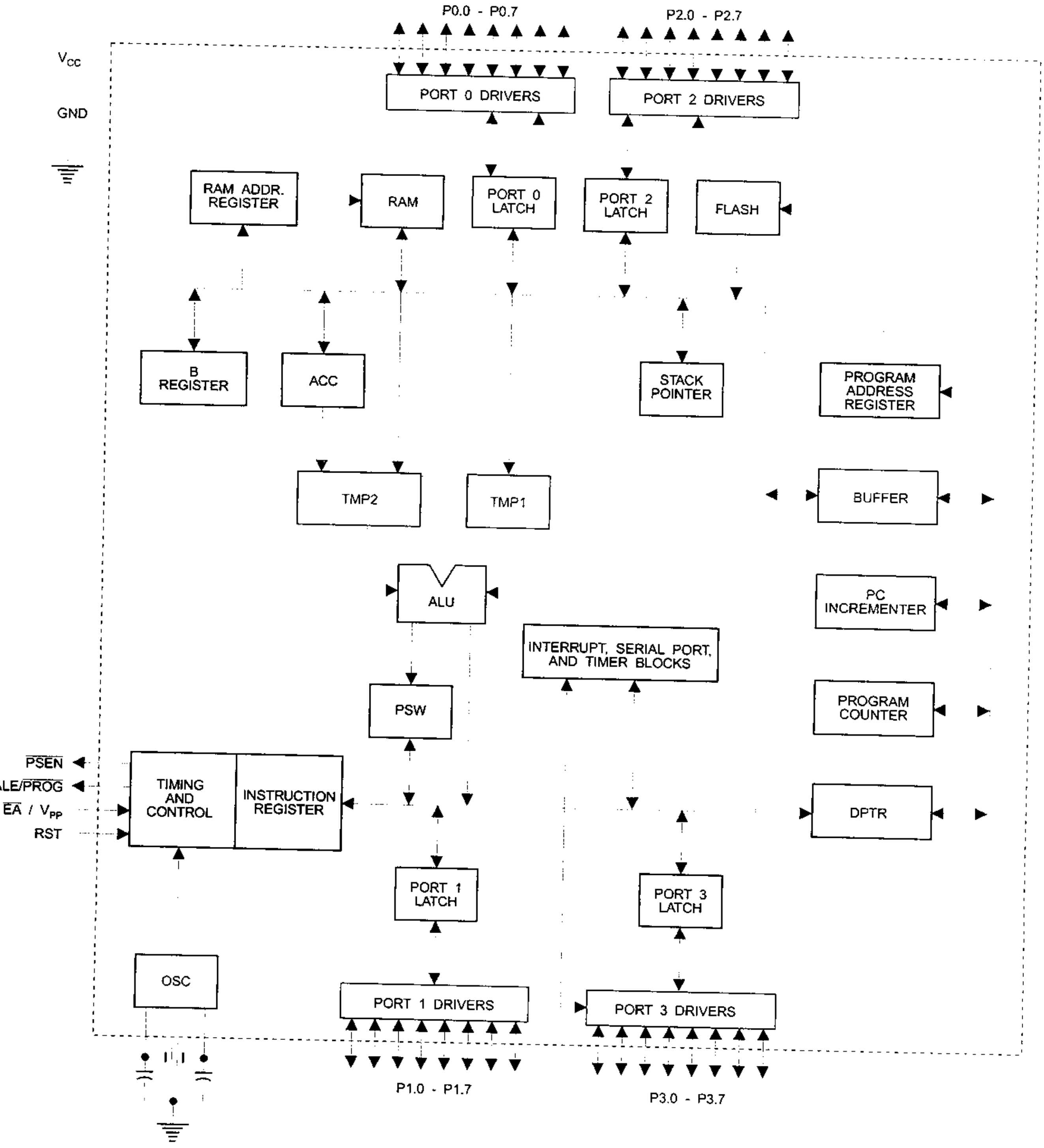
PQFP/TQFP



PLCC



lock Diagram



e AT89C51 provides the following standard features: 4K tes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit ner/counters, a five vector two-level interrupt architecture, full duplex serial port, on-chip oscillator and clock cir-itry. In addition, the AT89C51 is designed with static logic r operation down to zero frequency and supports two ftware selectable power saving modes. The Idle Mode ops the CPU while allowing the RAM, timer/counters, erial port and interrupt system to continue functioning. The ower Down Mode saves the RAM contents but freezes e oscillator disabling all other chip functions until the next ardware reset.

'in Description

cc
upply voltage.

ND
Ground.

ort 0
'ort 0 is an 8-bit open drain bidirectional I/O port. As an utput port each pin can sink eight TTL inputs. When 1s re written to port 0 pins, the pins can be used as high-npedance inputs.

'ort 0 may also be configured to be the multiplexed low-rder address/data bus during accesses to external pro-ram and data memory. In this mode P0 has internal pul-ups.

'ort 0 also receives the code bytes during Flash program-ning, and outputs the code bytes during program verifica-ion. External pullups are required during program verifica-ion.

Port 1
Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2
Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3
Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and verification.

RST
Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{\text{PROG}}$
Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{\text{PROG}}$) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{\text{PSEN}}$
Program Store Enable is the read strobe to external program memory.

When the AT89C51 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

EA/V$_{PP}$

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be nternally latched on reset.

EA should be strapped to V$_{CC}$ for internal program execuions.

This pin also receives the 12-volt programming enable voltge (V$_{PP}$) during Flash programming, for parts that require 2-volt V$_{PP}$.

TAL1

nput to the inverting oscillator amplifier and input to the nternal clock operating circuit.

TAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

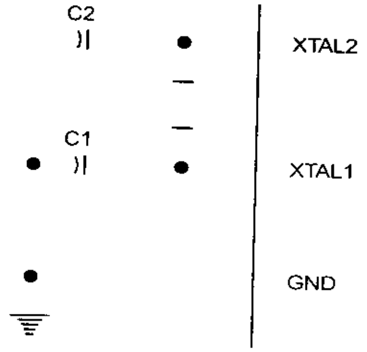
TAL1 and XTAL2 are the input and output, respectively, f an inverting amplifier which can be configured for use as n on-chip oscillator, as shown in Figure 1. Either a quartz rystal or ceramic resonator may be used. To drive the evice from an external clock source, XTAL2 should be left nconnected while XTAL1 is driven as shown in Figure 2. here are no requirements on the duty cycle of the external ock signal, since the input to the internal clocking circuitry through a divide-by-two flip-flop, but minimum and maxium voltage high and low time specifications must be bserved.

Ile Mode

idle mode, the CPU puts itself to sleep while all the onip peripherals remain active. The mode is invoked by ftware. The content of the on-chip RAM and all the speal functions registers remain unchanged during this ode. The idle mode can be terminated by any enabled errupt or by a hardware reset.

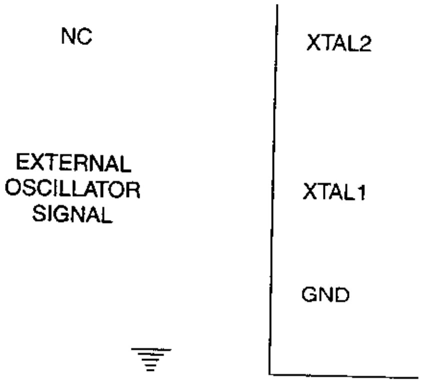
It should be noted that when idle is terminated by a hard ware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Figure 1. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



tatus of External Pins During Idle and Power Down Modes

ode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
le	Internal	1	1	Data	Data	Data	Data
le	External	1	1	Float	Data	Address	Data
ower Down	Internal	0	0	Data	Data	Data	Data
ower Down	External	0	0	Float	Data	Data	Data

Power Down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

When lock bit 1 is programmed, the logic level at the $\overline{\text{EA}}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{\text{EA}}$ be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

Program Lock Bits				Protection Type
	LB1	LB2	LB3	
1	U	U	U	No program lock features.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{\text{EA}}$ is sampled and latched on reset, and further programming of the Flash is disabled.
3	P	P	U	Same as mode 2, also verify is disabled.
4	P	P	P	Same as mode 3, also external execution is disabled.

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12V	V_{PP} = 5V
Top-Side Mark	AT89C51	AT89C51
	xxxx	xxxx-5
	yyww	yyww
Signature	(030H)=1EH	(030H)=1EH
	(031H)=51H	(031H)=51H
	(032H)=FFH	(032H)=05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figures 3 and 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{\text{EA}}$/V_{PP} to 12V for the high-voltage programming mode.
5. Pulse ALE/$\overline{\text{PROG}}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{\text{Data}}$ Polling: The AT89C51 features $\overline{\text{Data}}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{\text{Data}}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{\text{Busy}}$: The progress of byte programming can also be monitored by the RDY/$\overline{\text{BSY}}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

rogram Verify: If lock bits LB1 and LB2 have not been rogrammed, the programmed code data can be read back ia the address and data lines for verification. The lock bits annot be verified directly. Verification of the lock bits is chieved by observing that their features are enabled.

hip Erase: The entire Flash array is erased electrically y using the proper combination of control signals and by olding ALE/$\overline{\text{PROG}}$ low for 10 ms. The code array is written vith all "1"s. The chip erase operation must be executed efore the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are ead by the same procedure as a normal verification of ocations 030H,

031H, and 032H, except that P3.6 and P3.7 must be pulled o a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 51H indicates 89C51
(032H) = FFH indicates 12V programming
(032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode		RST	$\overline{\text{PSEN}}$	ALE/$\overline{\text{PROG}}$	$\overline{\text{EA}}$/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L	‾╲_╱‾	H/12V	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L	‾╲_╱‾	H/12V	H	H	H	H
	Bit - 2	H	L	‾╲_╱‾	H/12V	H	H	L	L
	Bit - 3	H	L	‾╲_╱‾	H/12V	H	L	H	L
Chip Erase		H	L	‾╲_╱‾ (1)	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10-ms PROG pulse.

HOLTEK

3^{18} Series of Encoders

Features

- Operating voltage: 2.4V~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

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 3^{18} encoders are a series of CMOS LSIs for remote control system applications. They are capable of encoding 18 bits of information which consists of N address bits and 18–N data bits. Each address/data input is externally trinary programmable if bonded out. It is otherwise set floating internally. Various packages of the 3^{18} encoders offer flexible combinations of programmable address/data to meet various application needs. The programmable address/data is transmitted together with the header bits via an RF or an infrared transmission medium upon receipt of a trigger signal. The capability to select a TE trigger type or a DATA trigger type further enhances the application flexibility of the 3^{18} series of encoders.

Selection Table

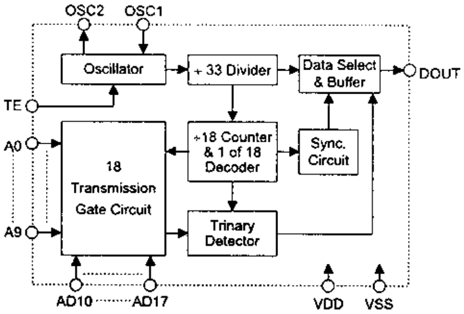
Function Part No.	Address No.	Address/ Data No.	Data No.	Dummy Code No.	Oscillator	Trigger	Package
HT600	9	5	0	4	RC oscillator	TE	20 DIP/20 SOP
HT640	10	8	0	0	RC oscillator	TE	24 SOP/24 SDIP
HT680	8	4	0	6	RC oscillator	TE	18 DIP
HT6187	9	0	3	6	RC oscillator	D12,D14,D15	18 DIP/20 SOP
HT6207	10	0	4	4	RC oscillator	D12~D15	20 DIP/20 SOP
HT6247	12	0	6	0	RC oscillator	D12~D17	24 SOP/24 SDIP

Note: Address/Data represents addressable pins or data according to the decoder requirements.

Block Diagram

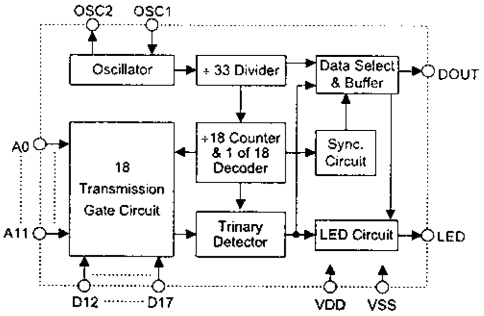
TE trigger

HT600/HT640/HT680



DATA trigger

HT6187/HT6207/HT6247

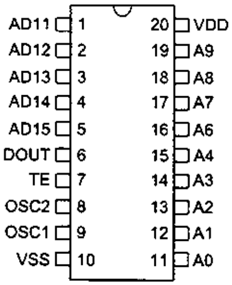


Note: The address/data pins are available in various combinations.

Pin Assignment

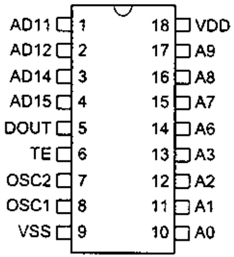
TE trigger type

9-Address
5-Address/Data



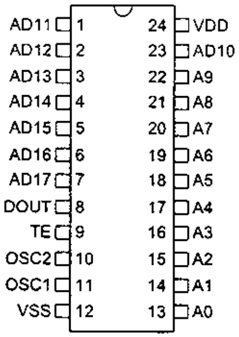
HT600
- 20 DIP/SOP

8-Address
4-Address/Data



HT680
- 18 DIP/SOP

10-Address
8-Address/Data



HT640
- 24 SOP/SDIP

HOLTEK

3^{18} Series of Decoders

Features

- Operating voltage: 2.4V~12V
- Low power and high noise immunity CMOS technology
- Low standby current
- Capable of decoding 18 bits of information
- Pairs with HOLTEK's 3^{18} series of encoders
- 8~18 address pins
- 0~8 data pins
- Trinary address setting
- Two times of receiving check
- Built-in oscillator needs only a 5% resistor
- Valid transmission indictor
- Easily interface with an RF or an infrared transmission medium
- Minimal external components

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 3^{18} decoders are a series of CMOS LSIs for remote control system applications. They are paired with the 3^{18} series of encoders. For proper operation a pair of encoder/decoder pair with the same number of address and data format should be selected (refer to the encoder/decoder cross reference tables).

The 3^{18} series of decoders receives serial address and data from that series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. It then compares the serial input data twice continuously with its local address. If no errors or unmatched codes are encountered, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission.

The 3^{18} decoders are capable of decoding 18 bits of information that consists of N bits of address and 18–N bits of data. To meet various applications they are arranged to provide a number of data pins whose range is from 0 to 8 and an address pin whose range is from 8 to 18. In addition, the 3^{18} decoders provide various combinations of address/data number in different packages.

Selection Table

Function / Item	Address No.	Data No.	Data Type	VT	Oscillator	Trigger	Package
HT602L	12	2	L	√	RC oscillator	DIN active "Hi"	20 DIP/20 SOP
HT604L	10	4	L	√	RC oscillator	DIN active "Hi"	20 DIP/20 SOP
HT605L	9	5	L	√	RC oscillator	DIN active "Hi"	20 DIP/20 SOP
HT611	14	0	—	√	RC oscillator	DIN active "Hi"	20 DIP/20 SOP

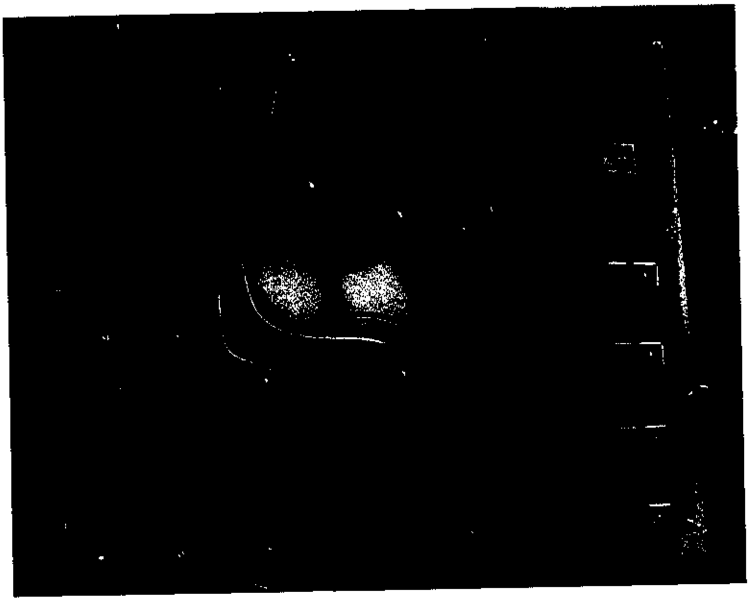
Function / Item	Address No.	Data No.	Data Type	VT	Oscillator	Trigger	Package
HT612	12	2	M	√	RC oscillator	DIN active "Hi"	20 DIP/20 SOP
HT614	10	4	M	√	RC oscillator	DIN active "Hi"	20 DIP/20 SOP
HT615	9	5	M	√	RC oscillator	DIN active "Hi"	20 DIP/20 SOP
HT644L	14	4	L	√	RC oscillator	DIN active "Hi"	24 SOP/24 SDIP
HT646L	12	6	L	√	RC oscillator	DIN active "Hi"	24 SOP/24 SDIP
HT648L	10	8	L	√	RC oscillator	DIN active "Hi"	24 SOP/24 SDIP
HT651	18	0	—	√	RC oscillator	DIN active "Hi"	24 SOP/24 SDIP
HT654	14	4	M	√	RC oscillator	DIN active "Hi"	24 SOP/24 SDIP
HT656	12	6	M	√	RC oscillator	DIN active "Hi"	24 SOP/24 SDIP
HT658	10	8	M	√	RC oscillator	DIN active "Hi"	24 SOP/24 SDIP
HT682L	10	2	L	√	RC oscillator	DIN active "Hi"	18 DIP
HT683L	9	3	L	√	RC oscillator	DIN active "Hi"	18 DIP
HT684L	8	4	L	√	RC oscillator	DIN active "Hi"	18 DIP
HT691	12	0	—	√	RC oscillator	DIN active "Hi"	18 DIP
HT692	10	2	M	√	RC oscillator	DIN active "Hi"	18 DIP
HT693	9	3	M	√	RC oscillator	DIN active "Hi"	18 DIP
HT694	8	4	M	√	RC oscillator	DIN active "Hi"	18 DIP

Note: Data type: M represents momentary type of data output.
L represents latch type of data output.

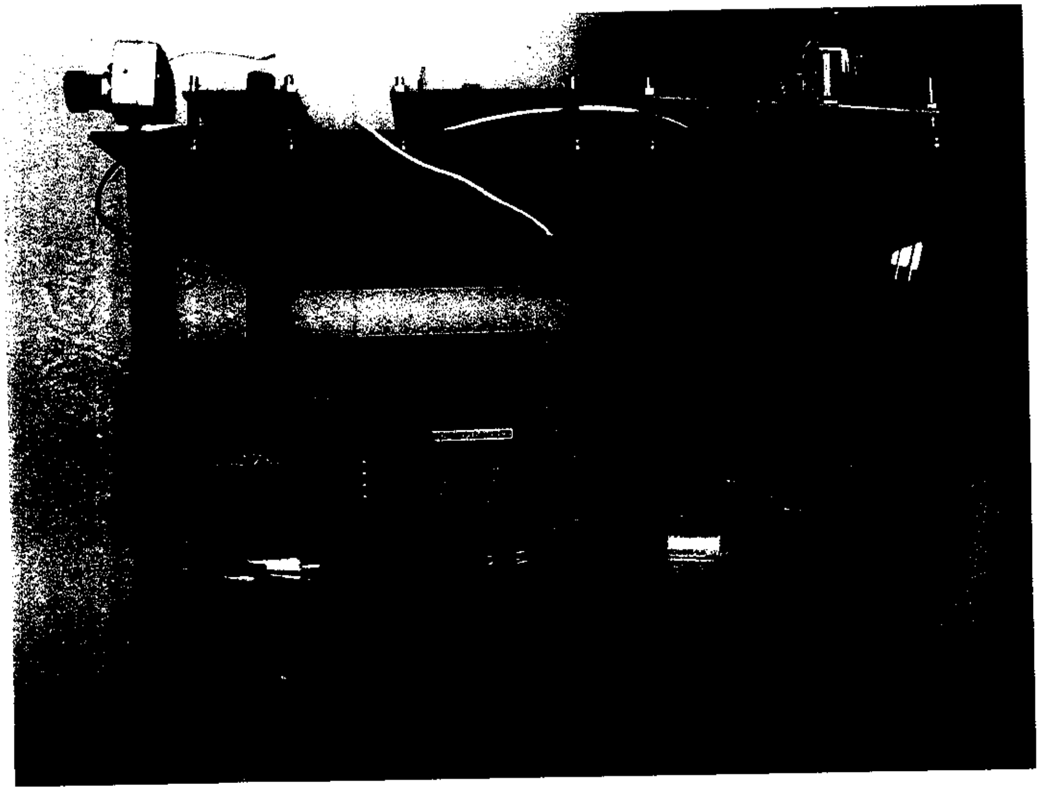
VT can be used as a momentary data output.

TRANSMITTR SECTION

RECEIVIER SECTION



MECHANICAL ASSEMBLY OF ROBOT SECTION



REFERENCES

1. An Article by Ayanna Howard, Senior Manager, NASA in IEEE Spectrum. March 2005

   Page no: 15
2. Ajay V. Deshmuk "Microcontroller Programming and Interfacing". Published by Tata McGRaw-Hill-New delhi.
3. www.rentron.com
4. www.atmel.com
5. http://www.robotics.com/robots.html
6. www.logix4u.net
7. www.kmitl.ac.th/~kswichit/ISP-Pgm3v0/ISP-Pgm3v0.html
8. http://www.8052.com/users/AT89S51InSystemProg/
9. http://www.esacademy.com/automation/docs/c51primer/c51prim.htm
10. http://www.faculty.ucr.edu/~currie/roboadam.htm
11. www.zen22142.zen.co.uk/Circuits/rf/tv.html