



P-1998



ACTUATING A REMOTE ROBOTIC ARM THROUGH A WIRELESS LINK

A PROJECT REPORT

Submitted by

SABARISH BAPU M.R.

71203106044

SAI SUBRAMANIAN.C

71203106045

VARUN.A

71203106056

VENKATA SUBRAMANIAM.R

71203106057

in partial fulfillment for the award of the degree

of

**BACHELOR OF ENGINEERING IN
ELECTRONICS AND COMMUNICATION
ENGINEERING**

**KUMARAGURU COLLEGE OF TECHNOLOGY,
COIMBATORE**

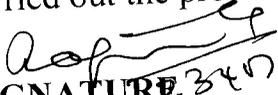
ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2007

ANNA UNIVERSITY: CHENNAI 600 025

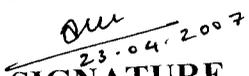
BONAFIDE CERTIFICATE

Certified that this project report "ACTUATING A REMOTE ROBOTIC ARM THROUGH A WIRELESS LINK" is the bonafide work of VARUN.A, SABARISH BAPU.M.R, SAI SUBRAMANIAN.C and VENKATA SUBRAMANIAM.R who carried out the project work under my supervision.


SIGNATURE

Dr. Rajeswari Mariappan

HEAD OF DEPARTMENT
Electronics & Communication
Engineering,
Kumaraguru College of
Technology,
Coimbatore - 641006

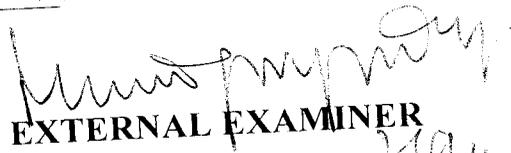

SIGNATURE

Ms.A.Vasuki

SUPERVISOR
Assistant Professor,
Electronics & Communication
Engineering,
Kumaraguru College of
Technology,
Coimbatore - 641006

The Candidates with University Register Nos. 71203106044, 71203106045, 71203106056, 71203106057 were examined by us in the Project Viva-voce examination held on 24.4.07.


INTERNAL EXAMINER
24/4/07


EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We are greatly indebted to our beloved Principal Dr. Joseph V. Thanikal, Ph.D., who has been the backbone of all our deeds.

We profusely thank Dr. Rajeswari Marippan, Ph.D., Head of Department, Department of Electronics and Communication Engineering, for lending a helping hand in this project.

We are highly grateful to our beloved Project Coordinator, Prof. K. Ramprakash, M.E., Professor and Project Guide Ms. A. Vasuki, M.E., Assistant Professor, ECE Department for their valuable guidance, timely help, constant encouragement and advice rendered throughout the project period for the successful completion of the same.

We owe much to all Lab Technicians of ECE department and Mechanical department for their support and motivation throughout the entire phase of the project in the development of the hardware and software.

We are grateful to all the faculty members of the Department of Electronics and Communication Engineering, who have helped us in innumerable ways.

We also thank our parents without whom we couldn't have come so far and friends for their timely help that culminated as good in the end.

ABSTRACT

The project "Actuating a Remote Robotic Arm through a Wireless Link" is about controlling a robotic arm with an object at the transmitting side. The robotic arm imitates the movements of the object (as per the arrangement) in the 2-D horizontal sweep. The controlling object is captured at two different positions. Angle between the two positions is calculated using motion detection algorithm after the patterns are recognized. The angular difference is encoded and transmitted to the robotic arm. The binary coded information is transferred through the serial port to the RF transmitter. The RF data is received and decoded. Then the data is fed to the serial port of the micro controller 8051. A stepper motor which can turn 1.8 degree per step is used for controlling the robotic arm. The stepper motor is connected to the parallel port of microcontroller 8051 through an interface board. The project is segregated into the following modules:

1. Image Processing
2. Encoding
3. Wireless Communication
4. Decoding
5. Arm Actuation

The project focuses on 2-D (horizontal) arm motion using a single stepper motor. This can be further enhanced to perform other movements as well as motions in 3-D in the future.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	LIST OF TABLES	x
	LIST OF FIGURES	xi
1	INTRODUCTION	1
1.1	CAMERA	1
1.2	IMAGE PROCESSING	2
1.3	VOLTAGE LEVEL CONVERSION	2
1.4	TRANSMITTER	2
1.5	RECEIVER	3
1.6	MICROCONTROLLER AND INTERFACING	3
1.7	STEPPER MOTOR	3
1.8	ROBOTIC ARM	3
2	IMAGE PROCESSING	4
2.1	INTRODUCTION	4
2.2	DIGITAL IMAGE REPRESENTATION	4
2.3	STEPS IN IMAGE PROCESSING	4
2.4	NEED FOR MATLAB	7
2.5	IMAGE PROCESSING IN MATLAB	7

2.5.1	IMAGE PROCESSING TOOLBOX	7
2.5.2	IMAGE REPRESENTATION IN MATLAB	7
2.5.3	IMAGE FORMATS SUPPORTED BY MATLAB	9
2.6	IMAGE ACQUISITION	9
2.6.1	WORKING WITH IMAGE ACQUISITION TOOLBOX	9
2.6.2	CONNECTING TO HARDWARE	10
2.6.3	PREVIEWING THE ACQUISITION	10
2.6.4	ACQUIRING IMAGE DATA	10
2.6.5	MANAGING IMAGE DATA TYPES	11
2.7	PATTERN RECOGNITION	12
2.7.1	MOMENTS	14
2.7.2	PATTERN RECOGNITION IN MATLAB	15
2.8	PIXEL TRACING FOR ANGLE CALCULATION	16
2.9	CREATING A SERIAL PORT OBJECT	17
2.9.1	SERIAL PORT DESCRIPTIVE PROPERTIES	18
2.9.2	SERIAL PORT OBJECT DISPLAY	18
2.9.3	SERIAL PORT PROPERTIES	19

	2.9.4	WRITING DATA IN SERIAL PORT	20
3		SERIAL COMMUNICATION	21
3.1		INTRODUCTION	21
3.2		ISSUES IN COMMUNICATION	21
	3.2.1	DATA TRANSMISSION	21
	3.2.2	PARALLEL TRANSMISSION	22
	3.2.3	SERIAL TRANSMISSION	22
3.3		MODES OF SERIAL DATA TRANSMISSION	23
	3.3.1	ASYNCHRONOUS TRANSMISSION	23
	3.3.2	SYNCHRONOUS TRANSMISSION	24
3.4		RS232 STANDARD	24
3.5		MAX 232	28
4		WIRELESS COMMUNICATION	31
4.1		INTRODUCTION	31
4.2		RF MODULE	31
	4.2.1	TRANSMITTER (TWS 434)	31
	4.2.2	RECEIVER (RWS434)	33
4.3		RF APPLICATIONS	35
4.4		SIGNAL CONVERSION	35

5	MICRO CONTROLLER AND INTERFACING	37
5.1	INTRODUCTION	37
5.2	MICRO CONTROLLER VS MICROPROCESSOR	37
5.3	NEED FOR MICROCONTROLLER	37
5.4	FEATURES OF 8051	38
	5.4.1 FEATURES	38
	5.4.2 8051 MICROCONTROLLER HARDWARE	39
	5.4.3 ARCHITECTURE OF 8051	39
5.5	SERIAL DATA INPUT/OUTPUT	40
5.6	INTERFACING TO STEPPER MOTOR	43
6	ARM ACTUATION	44
6.1	INTRODUCTION	44
6.2	STEPPER MOTOR	45
6.3	CHARACTERISTICS OF STEPPER MOTOR	46
6.4	UNIPOLAR STEPPER MOTOR	47
6.5	STEPPER MOTOR DRIVE CIRCUIT	47
6.6	APPLICATIONS OF STEPPER MOTOR	49
6.7	ARM MOVEMENT	50
7	CONCLUSION	51

APPENDIX 1 - 8051 MICROCONTROLLER	52
APPENDIX 2 – MAX232 DATA SHEETS	60
APPENDIX 3- POWER SUPPLY	62
APPENDIX 4- IMAGE PROCESSING ALGORITHM	67
APPENDIX 5-ARM ACTUATION ALGORITHM	69
REFERENCES	72

LIST OF TABLES

TABLE NO.	TITLE	PAGE NO.
2.1	SERIAL PORT PROPERTIES	18
2.2	SERIAL PORT OBJECT COMMUNICATION PROPERTIES	19
3.1	PIN CONFIGURATION IN RS 232 STANDARD	26
3.2	PIN CONFIGURATION OF MAX 232 IC	29
5.1	TIMER 1 BAUD RATES	43
6.1	EXCITATION OF STEPPER MOTOR	49

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
1.1	BASIC BLOCK DIAGRAM	1
2.1	STEPS IN IMAGE PROCESSING	6
2.2	IMAGE REPRESENTATION IN MATLAB	8
2.3	OBJECT BOUNDARY AND MOMENT PATTERN	14
3.1	PARALLEL TRANSMISSION	22
3.2	SERIAL TRANSMISSION	22
3.3	START & STOP BITS IN ASYNCHRONOUS SERIAL COMMUNICATION	23
3.4	SERIAL PORT OF PC	25
3.5	MAX 232 IC	28
3.6	MAX232 INTERFACING TO PC SERIAL PORT	30
4.1	TWS 434 PIN DIAGRAM	32
4.2	RWS 434 PIN DIAGRAM	33
4.3	INTERFACING RF RECIEVER TO MICRO CONTROLLER	36
5.1	ARCHITECTURE OF 8051	39
6.1	STEPPER MOTOR DRIVE CIRCUIT	48

1. INTRODUCTION

The project “Actuating a remote robotic arm through a wireless link” is about controlling a remote robotic arm that imitates the sweep actions performed at the transmitter side. The various modules involved in the project are as shown in Fig 1.1

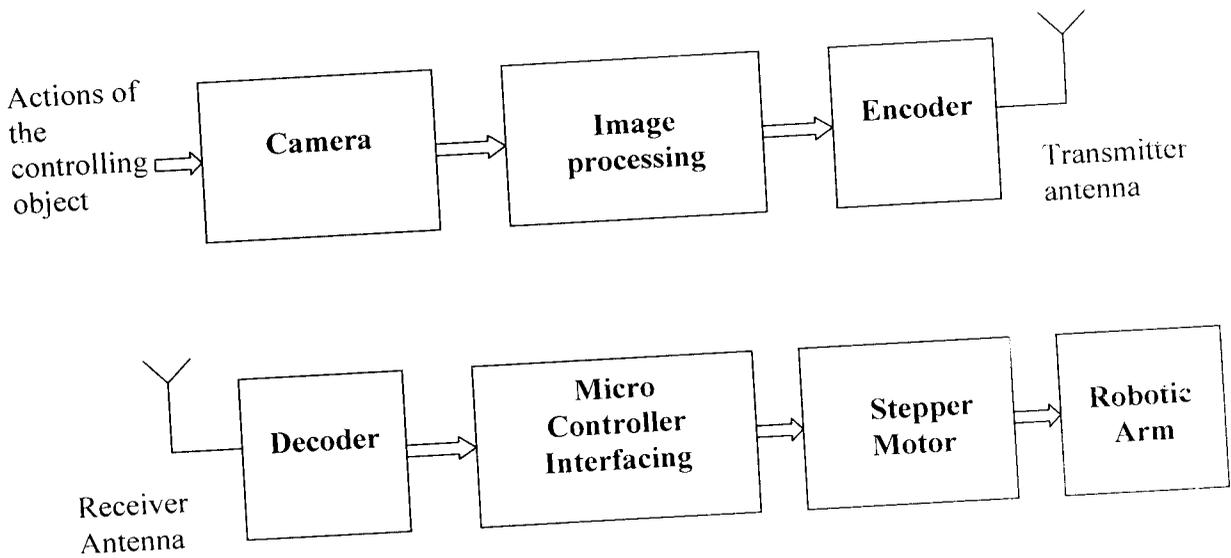


Fig 1.1 BASIC BLOCK DIAGRAM

1.1 CAMERA

The first step in the project is image acquisition. A web camera is used for this purpose. The resolution of the camera is 640*480 pixels. The controlling object is placed with a black background. The camera is finely adjusted to capture the position of the controlling object and is kept fixed in its position perpendicular to the plane in which the controlling object moves. It is connected to the computer and controlled by the software program.

1.2 IMAGE PROCESSING

The captured images are used to find the (spatial) angular movement of the controlling object which is kept in front of the camera. Image processing includes pattern recognition and motion detection sections. Once the patterns are recognized as the same, the angle is traced using motion detection algorithm. Pattern recognition is done using the concept of “Invariant moments”, which is reliable, independent of translation or rotation of the object. Angle calculation is performed with the help of pixel tracing. This algorithm gives the angular position of any specific color pixel in the captured image. The angle of the second position of the controlling object is to be transmitted. Encoding is taken care by the built-in features of Matlab.

1.3 VOLTAGE LEVEL CONVERSION

The encoded data from the serial port of computer will be in RS232 signal level which cannot be directly provided to transmitter. This signal is converted to TTL standard using a voltage converter circuit and given to the input of RF transmitter.

1.4 TRANSMITTER

Since the arm is located at a remote place the data is transmitted through a wireless link. RF transmitter is used for this purpose. RF transmitter comes in a versatile chip pack and is reliable. It operates on data signal in TTL logic level. The antenna can efficiently transmit the data for 100 meters.

1.5 RECIEVER

A RF receiver is used at the receiving side. It is an 8 pin IC which is as compact as the transmitter. Maximum operable rate is 6 kbps. The output of the receiver circuit is in TTL logic which is converted back to RS232 signal level before it is provided to the serial port of microcontroller.

1.6 MICROCONTROLLER AND INTERFACING

The (RS232 level) data is given to the serial port of the micro controller 8051. The micro controller receives the data at a pre-programmed baud rate (9600 or 4800 or 2400 or 1200) and stores the data in a memory location. Micro controller is programmed to find the difference angle with the available data of previous position. Now the micro controller excites the stepper motor. An interface board bridges the parallel port of the micro controller 8051 and the stepper motor by providing the necessary current gain and addressing logic.

1.7 STEPPER MOTOR

A DC stepper motor which can turn 1.8 degree per step with an average torque of 2 kilogram is used to actuate the arm. The stepper motor is controlled by the micro controller unit to turn clockwise for a positive angle and turn anti clockwise for a negative angle.

1.8 ROBOTIC ARM

The arm is connected to the rotor of the stepper motor. The weight of the arm is chosen within the operational capacity of the stepper motor. This arm arrangement moves in the direction in which the rotor turns. Thus the motion of the controlling object is imitated by the robotic arm.

2. IMAGE PROCESSING

2.1 INTRODUCTION

Interest in digital image processing stems from two principal application areas: improvement of pictorial information for human interpretation, and processing of scene data for autonomous machine perception. From 1964 until the present, the field of image processing has grown vigorously. In addition to applications in the space program, digital image processing techniques are now used to solve a variety of problems in the field of medicine, geography, archeology and so on.

2.2 DIGITAL IMAGE REPRESENTATION

The term image, refers to a two dimensional light intensity function $f(x, y)$, where x and y denote spatial coordinates and the value of f at any point (x, y) is proportional to the brightness (or gray level) of the image at that point. A digital image is an image $f(x, y)$ that has been discretized both in spatial coordinates and brightness. A digital image can be considered as a matrix whose row and column indices identify a point in the image and the corresponding matrix element value identifies the gray level at the point. The elements of such a digital array are called image elements, picture elements or pixels.

2.3 STEPS IN IMAGE PROCESSING

The first step in the process is image acquisition, i.e., to acquire a digital image. This requires an imaging sensor and the capability to digitize the signal produced by the sensor. The imaging sensor could be a color TV camera, line scan camera or web camera. If the output of the camera or other

imaging sensor is not already in digital form, an analog to digital converter digitizes it.

The next step deals with preprocessing the image. The key function of preprocessing is to improve the image in ways that increase the chances for success of the other processes.

The next stage deals with segmentation. It partitions an input image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. On the other hand, a rugged segmentation procedure brings the process a long way toward successful solution of an imaging problem. A weak or erratic segmentation algorithm results in failure. The output of the segmentation stage usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. The first decision that must be made is whether the data should be represented as a boundary or as a complete region. Boundary representation is appropriate when the focus is on external shape characteristics, such as corners and inflections. Regional representation is appropriate when the focus is on internal properties, such as texture or skeletal shape.

Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. A method must also be specified for describing the data so that features of interest are highlighted. Description, also called feature selection, deals with extracting features that are basic for differentiating one class of objects from another.

The last stage involves recognition and interpretation. Recognition is the process that assigns a label to an object based on the information provided by its descriptors. Interpretation involves assigning meaning to an ensemble of recognized objects.

Knowledge about a problem domain is coded into an image processing system in the form of knowledge database. This knowledge may be as simple as detailing regions of an image where the information of interest is to be located, thus limiting the search that has to be conducted in seeking that information.

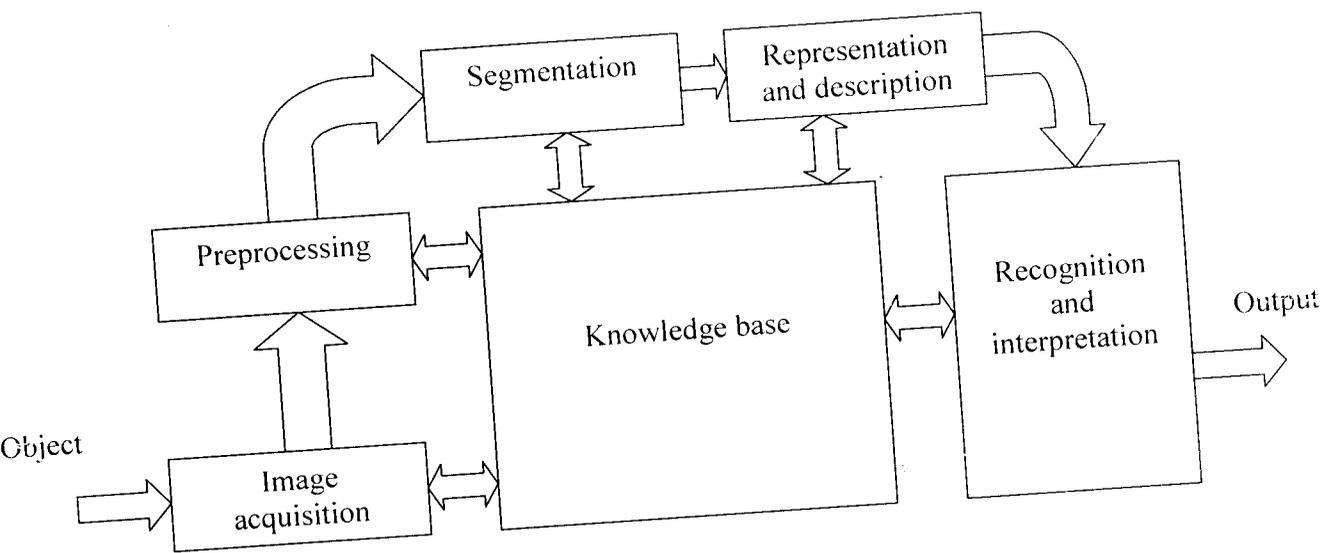


Fig 2.1 STEPS IN IMAGE PROCESSING

2.4 NEED FOR MATLAB

Image processing can be done using programming languages like 'C'. But it is tedious and extremely time consuming. The project makes use of Matlab for image processing. Image processing in Matlab is made easy to work with by the extensive supporting tools, toolboxes and pre-defined functions available. Moreover, any special function can be downloaded from Internet and can be made use of. Matlab supports image acquisition through 'imaq' toolbox and processing with 'image processing' toolbox. Acquisition camera and serial ports can be very easily controlled in terms of 'objects' concerned with their functions. Thus, Matlab makes the processes easy, reliable and tangible to the programmer's needs.

2.5 IMAGE PROCESSING IN MATLAB

2.5.1 IMAGE PROCESSING TOOLBOX

The Image Processing Toolbox extends the basic capabilities of Matlab by providing a number of specialized I/O, display, and processing functions for images and image processing.

2.5.2 IMAGE REPRESENTATION IN MATLAB

An image is stored as a matrix using standard Matlab matrix conventions. Ordinarily, image coordinates use the same conventions as matrix coordinates, with the first argument referring to row number and the second to column number. The origin is the upper left corner of the image. In Matlab, the origin is (1, 1). When Matlab asks for x and y coordinates, x is considered to be to the right and y is considered to be down.

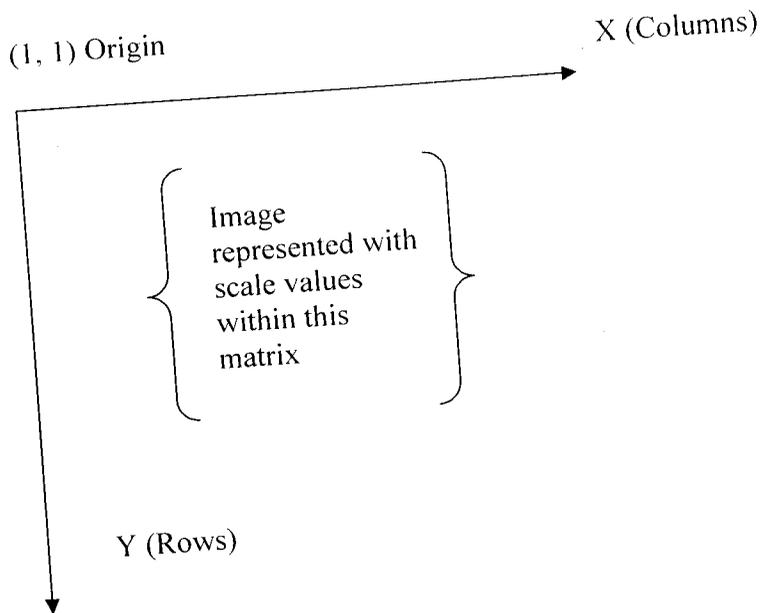


Fig. 2.2 IMAGE REPRESENTATION IN MATLAB

There are five basic types of images:

1. Intensity images
2. Binary images
3. Indexed images
4. RGB images
5. Multi frame images

➤ **Intensity image (gray scale image)**

This is the equivalent to a "gray scale image". It represents an image as a matrix where every element has a value corresponding to how bright/dark the pixel at the corresponding position should be colored. There are two ways to represent the number that represents the brightness of the pixel: The double class (or data type). This assigns a floating number ("a number with decimals") between 0 and 1 to each pixel. The value 0

corresponds to black and the value 1 corresponds to white. The other class is called `uint8` which assigns an integer between 0 and 255 to represent the brightness of a pixel. The value 0 corresponds to black and 255 to white. The class `uint8` only requires roughly 1/8 of the storage compared to the class `double`. On the other hand, many mathematical functions can only be applied to the `double` class. .

2.5.3 IMAGE FORMATS SUPPORTED BY MATLAB

The following image formats are supported by Matlab:

- BMP
- HDF
- JPEG
- PCX
- TIFF
- XWB

2.6 IMAGE ACQUISITION

2.6.1 WORKING WITH IMAGE ACQUISITION TOOLBOX

Image Acquisition Toolbox helps to connect and configure the hardware, preview the acquisition, and acquire and visualize image data. The toolbox functions can be used from the MATLAB command line and incorporate them into scripts (M-files) or into applications built within MATLAB.

2.6.2 CONNECTING TO HARDWARE

Image Acquisition Toolbox automatically detects compatible image acquisition devices from the MATLAB command line. The connection to the devices is encapsulated as an object, providing an interface for configuration and acquisition. Multiple connection objects can be created for simultaneous acquisition from as many devices as the PC and imaging hardware supports. The toolbox provides a consistent interface across multiple hardware devices and vendors, simplifying the configuration process. The hardware is configured by modifying the properties of the object associated with it. . The toolbox also supports camera files from hardware vendors.

Base properties can be set that are common to all supported hardware. These can include video format, resolution, region of interest (ROI), and returned color space. Device-specific properties can also be set such as hue, saturation, brightness, frame rate, contrast, video sync, and white and black reference levels.

2.6.3 PREVIEWING THE ACQUISITION

Image Acquisition Toolbox video preview window helps to verify and optimize the acquisition parameters. The preview window instantly reflects any adjustments that are made to acquisition properties.

2.6.4 ACQUIRING IMAGE DATA

Image Acquisition Toolbox can continuously acquire image data while the acquired data is processed in MATLAB. The toolbox automatically buffers acquired data into memory, handles memory and

buffer management, and enables acquisition from an ROI. It supports any color space provided by the image acquisition device, such as RGB, YUV, and grayscale. Raw sensor data in a Bayer pattern can be automatically converted into RGB data. The toolbox supports any frame rate and video resolution supported by the PC and imaging hardware.

2.6.5 MANAGING IMAGE DATA TYPES

Image Acquisition Toolbox represents image data in the MATLAB workspace as multidimensional arrays in any numerical data type supported by MATLAB, including uint8, uint16, uint32, single, and double. The image data is fully compatible with standard MATLAB functions and with Image Processing Toolbox. A typical image acquisition consists of the following four steps:

1. Creation -- creating a video input object
2. Configuration -- controlling acquisition behavior by setting object properties and previewing the results
3. Acquisition -- starting the video input object and acquiring and processing data
4. Clean-up -- deleting the video input object

Step 1: Creation

First, a video input object named vid is created using the video input function. The video input object is associated with a particular hardware device by passing the device's identification number to the function. For an example, the scientific video camera's device identification number is 1.

```
vid = videoinput('winvideo', 1);
```



Step 2: Configuration

Next, a preview window is opened.

```
preview(vid);
```

Step 3: Acquisition

The `getsnapshot` command is used to acquire a single video frame from the camera

```
data=getsnapshot(vid);
```

Step 4: Clean up

The video input object is deleted and cleared from the MATLAB workspace to free up memory.

```
delete(vid);
```

```
clear vid;
```

2.7 PATTERN RECOGNITION

A pattern is a quantitative or structural description of an object or some other entity of interest in an image. In general, a pattern is formed by one or more descriptors. In other words, a pattern is an arrangement of descriptors. A pattern class is a family of patterns that share common properties. Pattern recognition by machine involves techniques for assigning patterns to their respective classes automatically. The three principal pattern arrangements used in practice are vectors, strings and trees. Pattern vectors take the form as shown below.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

Where each component, x_i , represents the i th descriptor and n is the number of such descriptors. Pattern vectors are represented as columns (that is, $n \times 1$ matrices). The nature of the components of a pattern vector \mathbf{x} depends on the measurement technique used to describe the physical pattern itself.

There exist various algorithms for obtaining the pattern of any object. The efficiency of the algorithm used for finding the pattern depends on the effects of actions like object translation, rotation and movement on the resultant pattern. If the variance in the output pattern is higher for any of the above mentioned actions, then the algorithm fails to meet the requirement. Some of the algorithms for obtaining pattern are:

1. Chain codes
2. Polygonal approximation
3. Signatures
4. Boundary segments
5. Shape numbers
6. Fourier descriptors
7. Moments etc.

The project uses the concept of 'Invariant moments'.

2.7.1 MOMENTS

The shape of boundary of an object can be described quantitatively by using moments. The moments operate in such a way to compress the edge/segment of the object into a 1-D function in discrete domain. By this way, the complexity of representation decreases and is dependent on only one variable (say r , an arbitrary variable). Then, the output pattern can be considered as a mono dimensional function $g(r)$, which is a random variable.

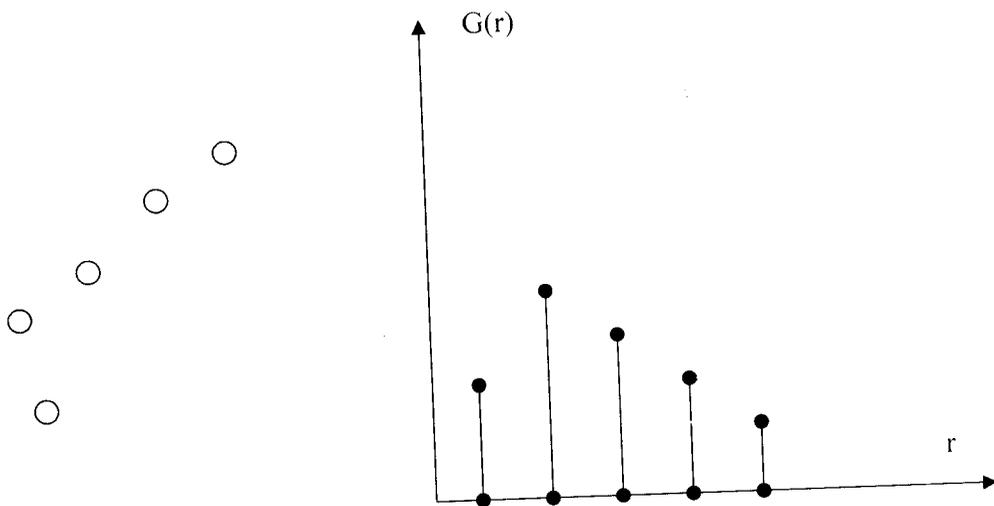


Fig. 2.3 OBJECT BOUNDARY AND MOMENT PATTERN

Basically, the operation done is to reduce the image description into a 1-D function. Although moments are far most popular method, they are not the only descriptors that could be used for this purpose. The advantage using moments over other techniques is that implementation of the other descriptors is that they carry a “physical” interpretation of the object which could be easily damaged by rotation or translation. Moments makes the representation

simple in a single dimension, so that it is very much immune to object rotation or translation or any other movement. This property helps to identify the controlling object motion in the project.

Moments are implemented in the project using the concept of 'Invariant moments' in Matlab within 'Dipum' toolbox. This toolbox is not available in direct packages. It should be downloaded from the Internet and made use of.

2.7.2 PATTERN RECOGNITION IN MATLAB

The pattern recognition in the project is used in order to avoid unwanted motions of the robotic arm for actions of an object other than the controlling object. This step makes sure that only the controlling object is mimicked.

In Matlab the two images captured is pattern recognized using the concept of 'Invariant moments'. Invariant moment is the functional implementation of moments in Matlab. When invariant moment is applied on a gray scale image, a matrix is generated corresponding to the object. This matrix is used for the comparison of the image patterns.

Two such matrices are generated for the images captured at two different positions. These two matrices are compared. If they are found to be the same then both the patterns are said to be the same. Only if the patterns are same angle calculated is utilized for the automation, else it is rejected. This step makes the system to react only for the actions performed by the controlling object. Any other unwanted or undesirable moving objects will not create any effect on the system. Invariant moments can be evoked as:

$C = \text{invmoments}(\text{Image})$

where,

C = resultant pattern which is a $1 \times N$ matrix in the workspace,

Image = the GRAY scale image whose pattern is found.

'invmoments' is the function called from DIPUM toolbox.

2.8 PIXEL TRACING FOR ANGLE CALCULATION

If the images captured are having the same pattern (if the images match) then the angle is calculated using pixel tracing. This angle is the information, which has to be transmitted to the arm.

A particular intensity image is chosen from the image and its co-ordinate (row-column) positions are found out. This is repeated for the second image also. With the obtained co-ordinates, spatial angles of the pixel are calculated.

The angle for each image is calculated using the co-ordinates. The position of second image is the crucial information and this angle is transferred to the serial port through a Matlab serial port object. Once the data is received, corresponding action of the control object is done using the stepper motor control program in micro controller. After each action, the currently received angular information used for updating 'initial' (ie., previous angular) position in the controller section. So this makes the transmission easier as only one angle is transmitted. The data is encoded before it is made available at the serial port.

2.9 CREATING A SERIAL PORT OBJECT

To create a serial port object in Matlab, the serial function is made use of. 'Serial' requires the name of the serial port connected to the computer device as an input argument and the property values during object creation can be configured. Each serial port object is associated with one serial port. For example, to create a serial port object associated with the COM1 port,

```
s = serial('COM1');
```

The serial port object *s* now exists in the MATLAB workspace. To display the class of *s* the 'whos' command can be used.

```
whos s
```

Name	Size	Bytes	Class
s	1x1	512	serial object

```
Grand total is 11 elements using 512 bytes
```

Once the serial port object is created, the properties listed below are automatically assigned values. These general purpose properties provide descriptive information about the serial port object based on the object type and the serial port.

2.9.1 SERIAL PORT DESCRIPTIVE PROPERTIES

Table 2.1 SERIAL PORT PROPERTIES

Property Name	Description
Name	specify a descriptive name for the serial port object
Port	Indicate the platform-specific serial name
Type	Indicate the object type

The values of these properties for `s` can be displayed with the `get` function.

```
get(s, {'Name', 'Port', 'Type'})
```

```
ans =
```

```
'Serial-COM1' 'COM1' 'serial'
```

2.9.2 SERIAL PORT OBJECT DISPLAY

The serial port object provides a convenient display that summarizes important configuration and state information. Summary information is displayed via the Workspace browser by right-clicking an instrument object and selecting Display Summary from the context menu.

The display summary for the serial port object s is given below.

Serial Port Object: Serial-COM1

Communication Settings

Port: COM1

Baud Rate: 9600

Terminator: 'LF'

Communication State

Status: closed

RecordStatus: off

Read/Write State

Transfer Status: idle

Bytes Available: 0

Values Received: 0

Values Sent: 0

2.9.3 SERIAL PORT PROPERTIES

Table 2.2 SERIAL PORT OBJECT COMMUNICATION PROPERTIES

Property Name	Description
Baud Rate	Specify the rate at which bits are transmitted.
Data Bits	Specify the number of data bits to transmit.
Parity	Specify the type of parity checking.
Stop Bits	Specify the number of bits used to indicate the end of a byte.

2.9.4 WRITING DATA IN SERIAL PORT

To send an integer data through serial port object, the following procedure is followed:

1. Creating a serial port object — The serial port object is created and is associated with serial port COM1.

```
s = serial('COM1');
```

2. Connecting to the instrument — Any serial port instrument connected to port COM1 can be accessed as:

```
fopen(s)
```

3. Writing data into the port:

To send the data, the command used is 'fwrite'. The syntax requires the serial port object name and the value being sent.

```
Fwrite(s,100)
```

4. Disconnect and clean up — When the serial port object is no longer needed, it should be disconnected from the instrument, and removed from memory (also from the MATLAB workspace). Command 'fclose', 'delete' and 'clear' are used for this purpose.

```
fclose(s)
```

```
delete(s)
```

```
clear s
```

Now a new serial object can be created with the port 'COM1'. Thus, the angle detail which is obtained from Matlab is taken out of the PC COM port. This data is converted to TTL logic and transmitted in wireless manner.

3. SERIAL COMMUNICATION

3.1 INTRODUCTION

Communication refers to meaningful exchange of information between the communicating entities. Therefore, in communication, it is concerned with all issues relating to exchange of information in the form of a dialog, e.g., dialog discipline, interpretation of messages, and acknowledgements.

3.2 ISSUES IN COMMUNICATION

1. Types of data communication
2. Modes of data transmission
3. Transmission Media
4. Serial Interface standards

3.2.1 DATA TRANSMISSION

There is always the need to exchange data, commands and other control information between a computer and its terminals, or between two computers. This information is in the form of bits. Data transmission refers to the movement of bits over some physical medium connecting two or more digital devices.

The two options of transmitting the bits are:

1. Parallel Communication
2. Serial Communication

3.2.2 PARALLEL TRANSMISSION

If more than 1 bit of information is transmitted over the data transmission medium at a time then it is considered as a parallel communication.

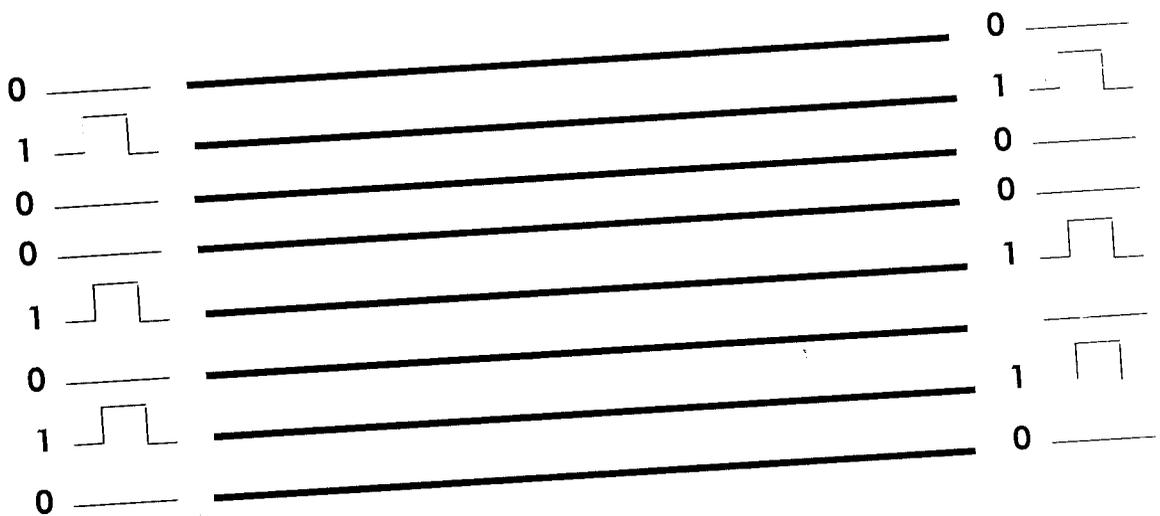


Fig 3.1 PARALLEL TRANSMISSION

3.2.3 SERIAL TRANSMISSION

The physical connection determines how many bits (1's or 0's) can be transmitted in a single instance of time.

If only 1 bit of information can be transmitted over the data transmission medium at a time then it is considered a serial communication.

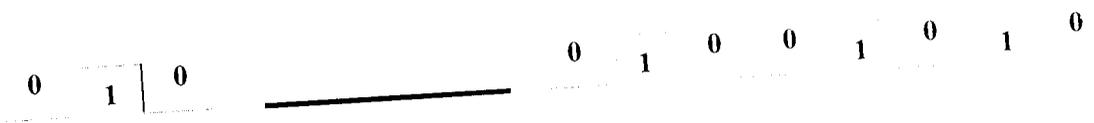


Fig 3.2 SERIAL TRANSMISSION

3.3 MODES OF SERIAL DATA TRANSMISSION

3.3.1 ASYNCHRONOUS TRANSMISSION

An action is called asynchronous when the agent performing an action does so whenever it wishes. Asynchronous transmission refers to the case when the sending Node commences transmission of bytes at any instant of time. Only one byte is sent at a time and there is no time relation between consecutive bytes, i.e., after sending a byte, the next byte can be sent after an arbitrary delay. Due to the arbitrary delay between consecutive bytes, the time occurrences of the clock pulses at the receiving end need to be synchronized repeatedly for each byte. This is usually achieved by providing two extra bits, a Start bit at the beginning and a Stop bit at the end of a byte.

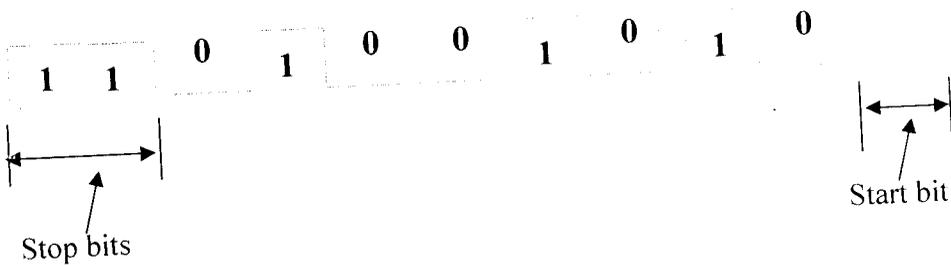


Fig 3.3 START & STOP BITS IN ASYNCHRONOUS SERIAL COMMUNICATION

3.3.2 SYNCHRONOUS SERIAL TRANSMISSION

A synchronous action, unlike an asynchronous action, is carried out under the control of a timing source. In synchronous transmission, bits are always synchronized to a reference clock irrespective of the bytes they belong to. There are no Start or Stop bits. Bytes are transmitted in a Block in a continuous stream of bits. Continuous transmission of bits enables the receiver to extract the clock from the incoming electrical signal.

3.4 RS-232 STANDARD

1. Most widely used serial i/o interfacing standard.
2. Used in PCs and numerous types of equipments.
3. It is not compatible with i/o voltage levels of TTL logic family

➤ VOLTAGE LEVELS IN RS232

1. Logic high(1) represented as -3 to -25V.
2. Logic low(0) represented as +3 to +25V.
3. -3 to +3v not defined.

➤ RS-232 INTERFACE

- RS-232 was introduced in 1960, and is currently the most widely used communication protocol. It is simple, inexpensive to implement, and though relatively slow; it is more than adequate for most simple serial communication devices such as keyboards and mice.

- RS-232 is a single-ended data transmission system, which means that it uses a single wire for data transmission.
- Since useful communication is generally two way, a two-wire system is employed, one to transmit and one to receive.

➤ SERIAL PORT OF PC

- IBM PC/compatible computers based on x86 (8086, 286, 386, 486 and Pentium) microprocessors normally have two COM (read: COMMUNICATION) ports. Both COM ports have RS – 232 type connectors. Many PCs use one each of the DB – 25 and DB – 9 RS – 232 connectors. The COM ports are designated as COM 1, COM 2 etc..... One can utilize COM X port of a PC for serial communication experiments, where X designates a particular port.

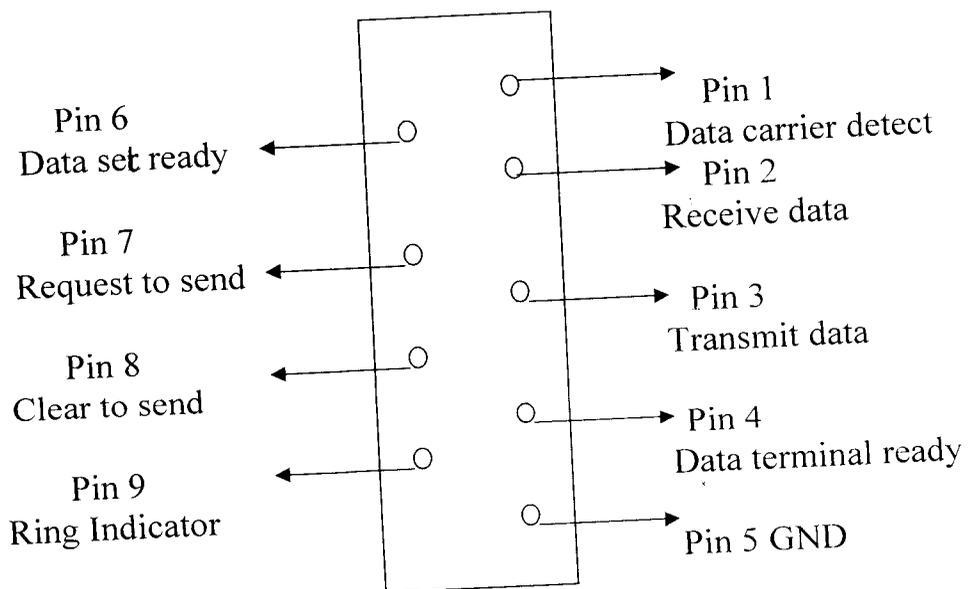


Fig 3.4 SERIAL PORT OF PC

Table 3.1 PIN CONFIGURATION IN RS232 STANDARD

PIN NO	SIGNAL
1	Carrier Detect (CD) (from DCE) Incoming signal from a modem
2	Received Data (RD) Incoming Data from a DCE
3	Transmitted Data (TD) outgoing Data to a DCE
4	Data Terminal Ready (DTR) Outgoing handshaking signal
5	Signal Ground Common reference voltage
6	Data Set Ready (DSR) Incoming handshaking signal
7	Request To Send (RTS) Outgoing flow control signal
8	Clear To Send (CTS) Incoming flow control signal
9	Ring Indicator (RI) (from DCE) Incoming signal from a modem

➤ **RS232 CONTROL LINES**

The RS232 standard describes the functions carried out by several control signals between the DTE and the DCE. The following control signals implement most of the important functions of an RS232 DTE to DCE link.

Data terminal ready (DTR)

This is a signal from the DTE to the DCE. When asserted, DTR indicates that the DTE is ready to accept data from the DCE. In systems with a modem, it maintains the connection and keeps the channel open. If DTR is negated, the communication path is broken. In everyday terms, negating DTR is the same as hanging up a phone.

Data set ready (DSR)

This is a signal from the DCE to the DTE, which indicates the readiness of the DCE. When this signal is asserted, the DCE is able to receive from the DTE. DSR indicates that the DCE (usually a modem) is switched on and is in its normal functioning mode (as opposed to its self-test mode).

Request to send

This is a signal from the DTE to the DCE. When asserted, RTS indicates to the DCE that the DTE wishes to transmit data to it.

Clear to send (CTS)

This is a signal from the DCE to the DTE and, when asserted, indicates that the DCE is ready to receive data from the DTE.

Rxd

It is used to receive the serial data transmitted from the external Microcontroller or circuit.

Txd

It is used to transmit the serial data to the external Micro controller or circuit from the PC.

➤ PC CONNECTION TO MAX

The RS232 standard is not TTL compatible; therefore, it requires the line driver such as MAX232 chip to convert the RS232 voltage levels to TTL levels and vice versa.

3.5 MAX232

RS232 does not use the conventional 0 and 5v implemented in TTL and CMOS designs. Drivers have to supply +5 to +15v for logic 0 and -5 to -15v for logical 1; this is performed by IC MAX232. This means that extra power supplies are needed to drive the RS232 voltage levels. Typically a +12 and a -12v power supply are used to drive the RS232 outputs. The first level charge pump essentially doubles the standard +5v power supply to provide the voltage level necessary for driving logic 0. A second charge pump inverts this voltage and provides the voltage level necessary for driving logic 1. These two charge pumps allow the RS232 interface products to operate from a single +5v supply.

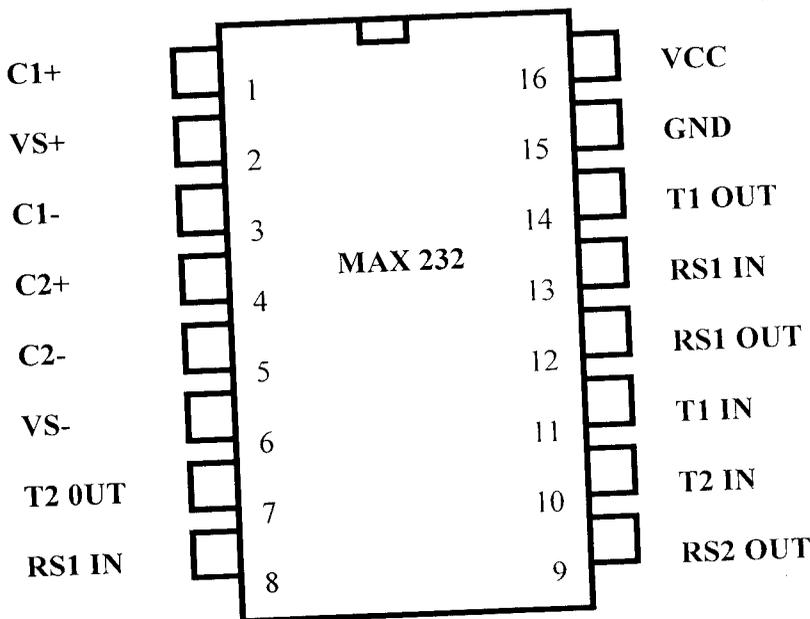


Fig 3.5 MAX 232 IC

Table 3.2 PIN CONFIGURATION OF MAX232 IC

MAX232 DIP Package Pin Layout

No.	Name	Purpose	Signal Voltage
1	C1+	+ connector for capacitor C1	capacitor should stand at least 16V
2	V+	output of voltage pump	+10V
3	C1-	- connector for capacitor C1	capacitor should stand at least 16V
4	C2+	+ connector for capacitor C2	capacitor should stand at least 16V
5	C2-	- connector for capacitor C2	capacitor should stand at least 16V
6	V-	output of voltage pump / inverter	-10V
7	T2out	Driver 2 output	RS-232
8	R2in	Receiver 2 input	RS-232
9	R2out	Receiver 2 output	TTL
10	T2in	Driver 2 input	TTL
11	T1in	Driver 1 input	TTL
12	R1out	Receiver 1 output	TTL
13	R1in	Receiver 1 input	RS-232
14	T1out	Driver 1 output	RS-232
15	GND	Ground	0V
16	VCC	Power supply	+5V

➤ MAX 232 INTERFACING

The data from Matlab is made available to the serial port. The encoded RS 232 level signal from pin 3 (TX) serial port is given to pin 8 (IN RS) of MAX 232 IC. The TTL level signal is obtained across pin 9 (OUT TTL) of the IC and is directly given to the RF transmitter. Thus, with this signal level conversion, the data is now ready to be transmitted.

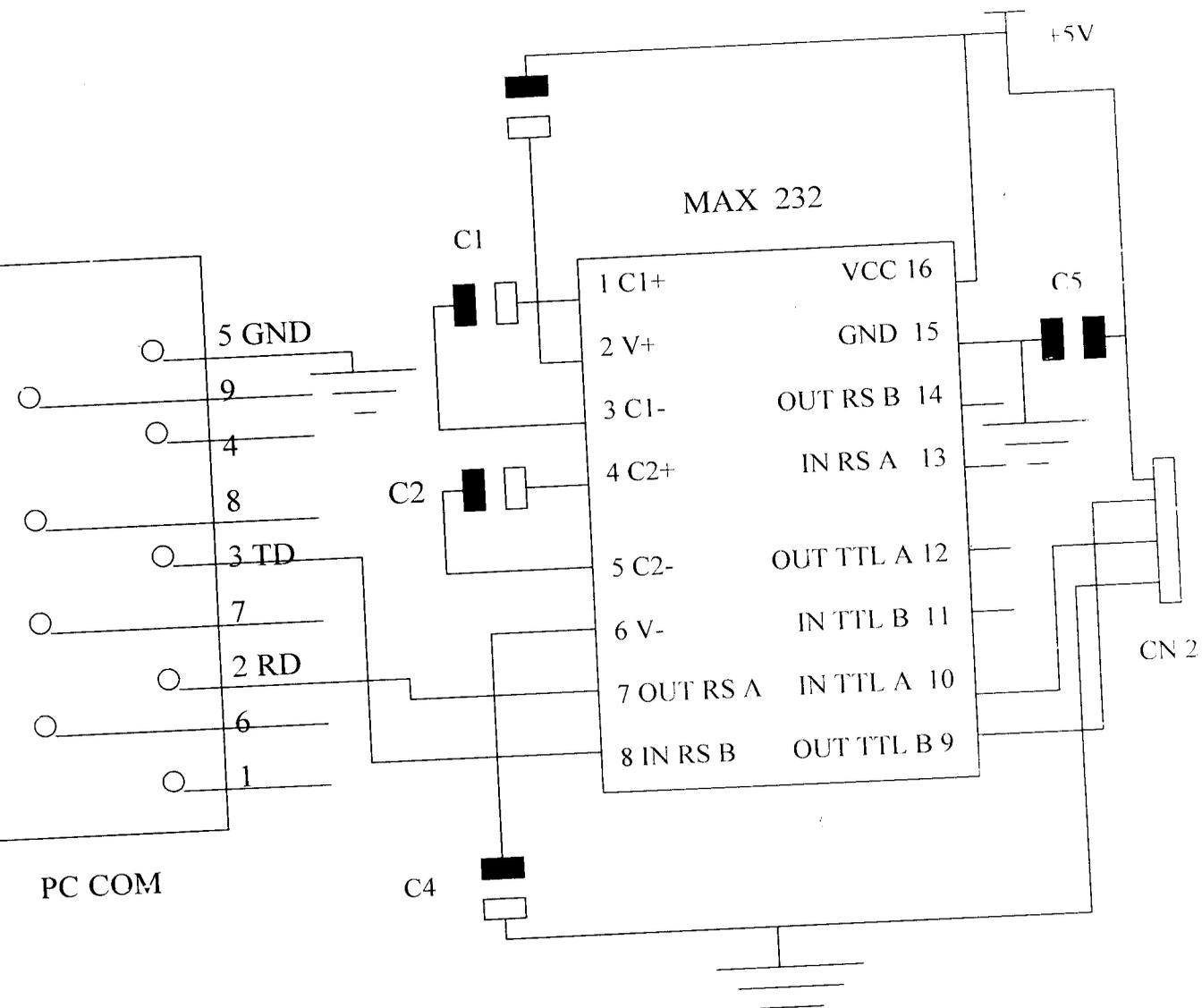


Fig 3.6 MAX 232 INTERFACED TO PC SERIAL PORT

4. WIRELESS COMMUNICATION

4.1 INTRODUCTION

The project makes use of wireless transmission in order to improve the range (distance) of operation. As wireless communication is used, the receiver can be kept at a remote place (here, a finite distance) and can be controlled. Wireless section gives an added advantage to the project and makes it applicable for any automation practice.

4.2 RF MODULES

RF module consists of:

1. Transmitter section (TWS 434) and
2. Receiver section (RWS 434)

The TWS-434 and RWS-434 are extremely small and are excellent for applications requiring short-range RF remote controls. These modules are selected because of their reliable data transmission capability blended with good data rates.

4.2.1 TRANSMITTER (TWS 434)

The transmitter module is only 1/3 of the size of a standard postage stamp and can easily be placed inside a small plastic enclosure. The transmitter output is up to 8mW at 433.92MHz with a range of approximately 400 foot (open area) for outdoors and the range is approximately 200 foot for indoors.

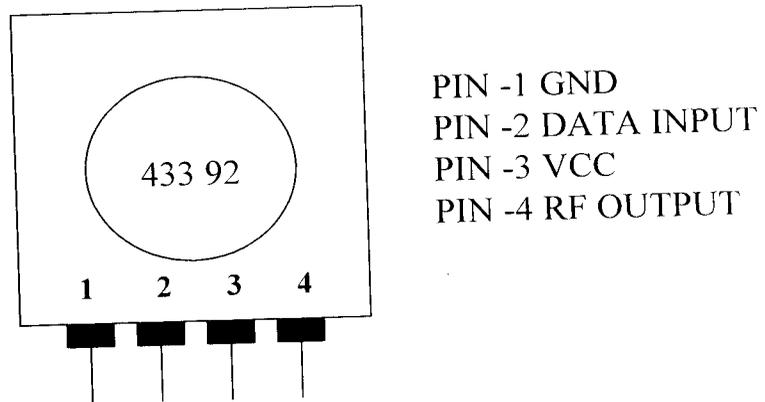


Fig 4.1 TWS-434 PIN DIAGRAM

Transmitter package consists of 4 pins whose operations are

1. Gnd

Ground of transmitter. This pin is to be connected to the common ground along with the MAX232 IC interfacing.

2. Data Input

The voltage level converted (RS232 to TTL logic) signal is given as the input to the transmitter. This signal is available at the pin 9 (OUT TTL) of the MAX232 interface circuit.

3. Supply

The operating voltage of the transmitter can be fixed between 2-12 dc volts. This varied range of supply makes the transmitter workable in a variety of input signal range.

4. Antenna

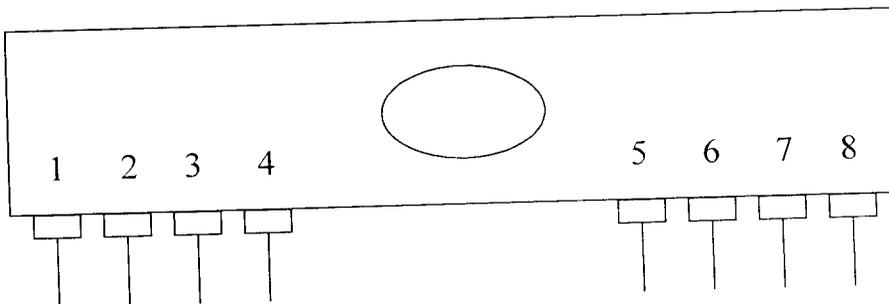
The transmitter can be connected to an antenna to improve the output quality and distance of operation. A normal whip antenna could make the transmitter power to be available up to 100meters in closed space environment.

➤ FEATURES OF TWS 434

- The TWS-434 transmitter accepts both linear and digital inputs
- It can operate from 1.5 to 12 Volts-DC
- Building a miniature hand-held RF transmitter is very easy here.

4.2.2 RECEIVER (RWS434)

The receiver also operates at 433.92MHz, and has a sensitivity of 3uV. It is a compact 8 pin version which is placed in a location within the operating range of the transmitter. The pin configuration is as:



- 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

Fig 4.2 RWS 434 PIN LAYOUT

Pin configuration of RWS434 is

1. Gnd

Pins 1, 6 and 7 are configured as the ground for the receiver. These pins are to be connected in common with the voltage conversion circuit which follows the receiver part.

2. Supply

Pins 4,5 are used for supply. The advantage of using this IC receiver is that the voltage range of operation (4.5 to 5.5V) is same as the operating voltage for MAX232 IC (which is to be used before micro controller section). This makes a reduction in use of separate supplies for both the circuits and makes the project less bulky.

3. Output

Both digital and analog output versions are supported by this IC. Digital output from pin 2 is the concern of the project. This signal which is in TTL level is to be converted to RS232 level before fed into the serial port of the micro controller.

4. Antenna

An antenna can be connected to the pin 8 of the receiver. Yet, for small distance reception, the antenna is optional. Receiver is reliable even without the usage of an antenna. The dimension of the antenna can be around 30-35 cm if used.

➤ **FEATURES OF RWS434**

- The RWS-434 receiver operates from 4.5 to 5.5 volts-DC.
- It has both linear and digital outputs.
- The receiver has channel spacing of 150 KHz and data rate of 4800 bps. Frequency Range:433.92MHz
- Modulate Mode: ASK
- Sensitivity: -119 dB
- High Sensitivity Passive Design
- Simple To Apply with Low External Count

4.3 RF APPLICATIONS

- Wireless security systems
- Car alarms
- Remote gate controls
- Remote sensing
- Data communication
- Remote water/electric switch
- Pager system
- PDA data communication

4.4 SIGNAL CONVERSION

The signals obtained from the receiver are in the TTL logic levels. In order to convert them as RS232 signal, a MAX232 interface circuit is designed. This circuit is very similar to the one used in transmitter side, with a small change of input and output data conversion.

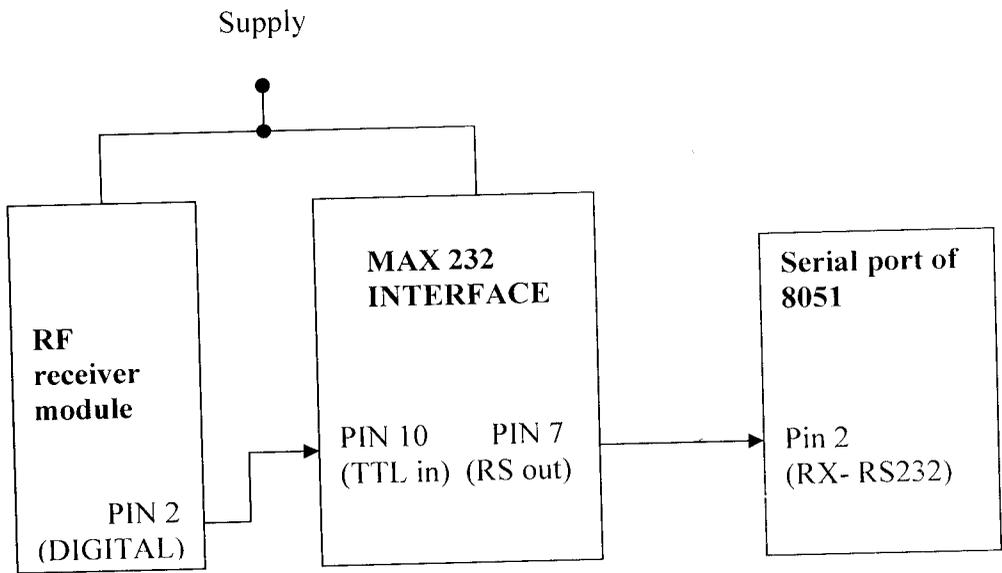


Fig 4.3 INTERFACING RF RECEIVER TO MICRO CONTROLLER

5 MICRO CONTROLLER AND INTERFACING

5.1 INTRODUCTION

Micro controllers help in every automation and controlling applications. The project uses micro controller 8051 which is very versatile for programming and fixes exactly for the need. The important functions of the micro controller in the project are to:

1. Recieve serial data (in a pre- programmed baud rate)
2. Actuate the stepper motor (like wise, the arm) with the data available.

Micro controller can be considered as the crucial segment of the project in the receiver section on which the complete automation is dependent upon.

5.2 MICROCONTROLLER VS MICROPROCESSOR

Microprocessor means the general purpose microprocessors such as Intel's X 86 family or Motorola's 680X0 families. The microprocessors contain no RAM, no ROM, and no I/O Ports on the chip itself. The microprocessors such as Pentium or the 68040 must add RAM, ROM, I/O Ports and timers externally to make them functional. Additional external RAM, ROM and Timers cause the system bulkier and more expensive.

5.3 NEED FOR MICROCONTROLLER

A microcontroller has a CPU in addition to a fixed amount of RAM, ROM and I/O ports all on single chip. In other words, the processor, the RAM, ROM, I/O ports and a timer and all embedded together in a single chip, therefore the designer no need to add external

RAM, ROM and timer etc. cost reduced and complexity of design the circuit also reduced. Microcontrollers are widely used in the embedded system products to perform specified task.

5.4 FEATURES OF 8051

5.4.1 FEATURES

- 4-Kbytes of on-chip Reprogrammable Flash Memory
- 128 x 8 RAM
- Two 16-bit Timer/Counters
- Full duplex serial channel
- Boolean processor
- Four 8-bit I/O ports, 32 I/O lines
- Memory addressing capability
- 64K ROM and 64K RAM
- Program memory lock
- Power save modes:
 - Idle and power-down
- Six interrupt sources
- Most instructions execute in 0.3 ms
- CMOS and TTL compatible
- Maximum speed: 40 MHz @ $V_{cc} = 5V$
- Packages available:
 - 40-pin DIP
 - 44-pin PLCC

5.4.2 8051 MICROCONTROLLER HARDWARE

The detail descriptions of the AT89C51 included are:

- Memory Map and Registers
- Timer/Counters
- Serial Interface
- Interrupt System
- Other Information
- Flash Memory

5.4.3 ARCHITECTURE OF 8051

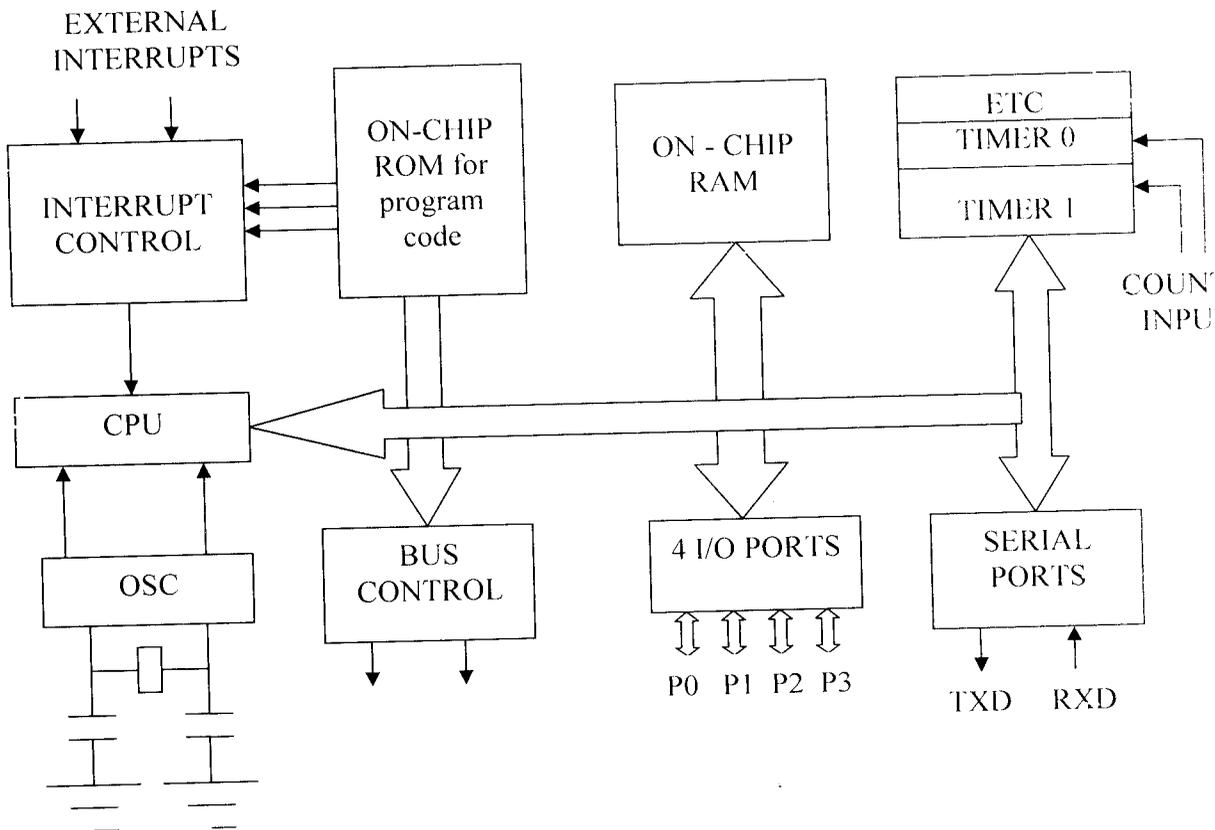


Fig 5.1 ARCHITECTURE OF 8051

5.5 SERIAL DATA INPUT/OUTPUT

The serial port is full duplex, which means it can transmit and receive simultaneously. It is also receive-buffered, which means it can begin receiving a second byte before a previously received byte has been read from the receive register. (However, if the first byte still has not been read when reception of the second byte is complete, one of the bytes will be lost.) The serial port receive and transmit registers are both accessed at Special Function Register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register. The serial port can operate in the following four modes.

SERIAL PORT CONTROL REGISTER

The serial port control and status register is the Special Function Register SCON. This register contains the mode selection bits, the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (TI and RI).

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

SM0	SCON.7	Serial port mode specifier
SM1	SCON.6	Serial port mode specifier
SM2	SCON.5	Used for multiprocessor comm.(make it 0)
REN	SCON.4	Enable/disable reception
TB8	SCON.3	Not widely used
RB8	SCON.2	Not widely used
TI	SCON.1	Transmit interrupt flag
RI	SCON.0	Receive interrupt flag

1. Mode 0

Serial data enters and exits through RXD. TXD outputs the shift clock. Eight data bits are transmitted/ received, with the LSB first. The baud rate is fixed at 1/12 the oscillator frequency.

2. Mode 1

10 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

3. Mode 2

11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable ninth data bit, and a stop bit (1). On transmit, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) can be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.

4. Mode 3

11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable ninth data bit, and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except the baud rate, which is variable in Mode 3. In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition $RI = 0$ and $REN = 1$. Reception is initiated in the other modes by the incoming start bit if $REN = 1$.

BAUD RATES

The baud rate in Mode 0 is fixed as shown in the following equation. The baud rate in Mode 2 depends on the value of the SMOD bit in Special Function Register PCON. If SMOD = 0 (the value on reset), the baud rate is 1/64 of the oscillator frequency. If SMOD = 1, the baud rate is 1/32 of the oscillator frequency, as shown in the following equation. In the AT89C51, the Timer 1 overflow rate determines the baud rates in Modes 1 and 3. In the AT89C52, these baud rates can be determined by Timer 1, by Timer 2, or by both (one for transmit and the other for receive).

USING TIMER1 TO GENERATE BAUD RATES

When Timer 1 is the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD according to the following equation. The Timer 1 interrupt should be disabled in this application. The Timer itself can be configured for either timer or counter operation in any of its 3 running modes. In the most typical applications, it is configured for timer operation in auto-reload mode (high nibble of TMOD = 0010B). In this case, the baud rate is given by the following formula. Programmers can achieve very low baud rates with Timer 1 by leaving the Timer 1 interrupt enabled, configuring the Timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 interrupt to do a 16-bit software reload. Lists of commonly used baud rates and how they can be obtained from Timer 1.

Table 5.1 TIMER 1 BAUD RATES

Baud rate	TH1(Decimal)	TH1(Hex)
9600	-3	FD
4800	-6	FA
2400	-12	F4
1200	-24	E8

MODE 0 BAUD RATE

Baud rate = $(2^{\text{SMOD}} * \text{Oscillator Frequency}) / (32 * 12 * [256 - \text{TH1}])$.

Mode 1 baud rate:

Baud rate = $(1/32) * \text{Osc Freq}$ when SMOD=1.

Baud rate = $(1/64) * \text{Osc Freq}$ when SMOD=0.

5.6 INTERFACING TO STEPPER MOTOR

The micro controller is connected to the stepper motor through an interface board. This circuit provides the necessary addressing logic and drives the stepper motor. Addressing is done using D-latches (74175) and the necessary current gain for exciting the coils of the stepper motor is provided by 4 Darlington pair of transistors. These transistors are directly connected to the coils. The voltage output at the board pin for the stepper motor is in the range of 12V per each coil. The specific coil is excited by the control coding which is executed in the micro controller.

6. ARM ACTUATION

6.1 INTRODUCTION

Stepper motors behave differently from standard DC motors. First of all, they cannot run freely by themselves. Stepper motors do as their name suggests -- they "step" a little bit at a time. Stepper motors also differ from DC motors in their torque-speed relationship. DC motors generally are not very good at producing high torque at low speeds, without the aid of a gearing mechanism. Stepper motors, on the other hand, work in the opposite manner. They produce the highest torque at low speeds.

Stepper motors also have another characteristic, 'holding torque', which is not present in DC motors. Holding torque allows a stepper motor to hold its position firmly when not turning. This can be useful for applications where the motor may be starting and stopping, while the force acting against the motor remains present. This eliminates the need for a mechanical brake mechanism. Steppers don't simply respond to a clock signal, they have several windings which need to be energized in the correct sequence before the motor's shaft will rotate. Reversing the order of the sequence will cause the motor to rotate the other way.

If the control signals are not sent in the correct order, the motor will not turn properly. It may simply buzz and not move, or it may actually turn, but in a rough or jerky manner. A circuit which is responsible for converting step and direction signals into winding energisation patterns is called a translator. Most stepper motor control systems include a driver in addition to the translator, to handle the current drawn by the motor's windings.

6.2 STEPPER MOTOR

A stepper motor is defined in British standard as “ a reversible brushless DC motor” the rotor of which rotates in discrete angular steps when its stator windings reenergized in a programmed manner, rotation occurs as a result of interaction between the rotor poles of the sequentially energized stator windings. The rotor carries no electrical windings but rather has salient soft or magnetized poles. Stepper motor of three basic types

1. **Permanent magnet (PM)** - utilizes a rotor having permanently magnetized salient poles of hard magnetic material.
2. **Variable reluctance (VR)** - utilizes a rotor having soft magnetic material
3. **Hybrid (HY)** - utilizes a rotor comprising an axial permanent magnet juxtaposed two soft iron rotor discs having salient poles.

In a stepper motor, the task of energizing the stator winding in the prescribed sequence is performed by an electronic controller in response to pulses received from an exogenous source, such as a microcomputer or a programmed tape or disc. The stepper motor is thus an incremental actuator, which receives input in the form of pulses, and whose output consequently is the actuator of choice in computer control systems.

A basic of the "translator + driver" type of configuration notice the separate voltages for logic and for the stepper motor. Usually the motor will require a different voltage than the logic portion of the system. Typically logic voltage is $+5 V_{dc}$ and the stepper motor voltage can range from $+5 V_{dc}$ up to about $+48 V_{dc}$. The driver is also an "open collector" driver, wherein it takes its outputs to GND to activate the motor's windings. Most semiconductor circuits are more capable of sinking (providing a GND or negative voltage) than sourcing (outputting a positive voltage).

6.3 CHARACTERISTICS OF STEPPER MOTOR

Stepper motors are not just rated by voltage. The following elements characterize a given stepper motor:

VOLTAGE

Stepper motors have a voltage rating. This is either printed directly on the unit, or is specified in the motor's datasheet. Exceeding the rated voltage is sometimes to obtain the desired torque from a given motor, but doing so may produce excessive heat & shorten the life of the motor.

RESISTANCE

Resistance-per-winding is another characteristic of a stepper motor. This resistance will determine current draw of the motor, as well as affect the motor's torque curve and maximum operating speed.

DEGREES PER STEP

This is often the most important factor in choosing a stepper motor for a given application. This factor specifies the number of degrees the shaft will rotate for each full step. Half step operation of the motor will double the number of steps/revolution, and cut the degrees-per-step in half. For unmarked motors, it is often possible to carefully count, by hand, the number of steps per revolution of the motor. The degrees per step can be calculated by dividing 360 by the number of steps in 1 complete revolution. Common degree/step numbers include: 0.72, 1.8, 3.6, 7.5, 15, and even 90. Degrees per step are often referred to as the resolution of the motor. As in the case of an unmarked motor, if a motor has only the number of steps/revolution printed on it, dividing 360 by this number will yield the degree/step value.

6.4 UNIPOLAR STEPPER MOTOR

Unipolar motors are relatively easy to control. A simple 1-of-'n' counter circuit can generate the proper stepping sequence, and drivers as simple as 1 transistor per winding are possible with unipolar motors. Unipolar stepper motors are characterized by their center-tapped windings. A common wiring scheme is to take all the taps of the center-tapped windings and feed them +MV (Motor voltage). The driver circuit would then ground each winding to energize it. Unipolar stepper motors are recognized by their center-tapped windings. The number of phases is twice the number of coils, since each coil is divided in two. In addition to the standard drive sequence, high-torque and half-step drive sequences are also possible.

In the high-torque sequence, two windings are active at a time for each motor step. This two-winding combination yields around 1.5 times more torque than the standard sequence, but it draws twice the current. Half-stepping is achieved by combining the two sequences. First, one of the windings is activated, then two, then one, etc. This effectively doubles the number of steps the motor will advance for each revolution of the shaft, and it cuts the number of degrees per step in half.

6.5 STEPPER MOTOR DRIVE CIRCUIT

A unique type of motor useful for moving things in small increments is the stepper motor. Instead of rotating smoothly around as most motors do, stepper motors rotate or step from one fixed position to the next. Common step sizes range from 0.9 to 30 degrees.

A stepper motor is stepped from one position to the next by changing the currents through the fields in the motor. The two common field connections are referred to as two phases and four phases.

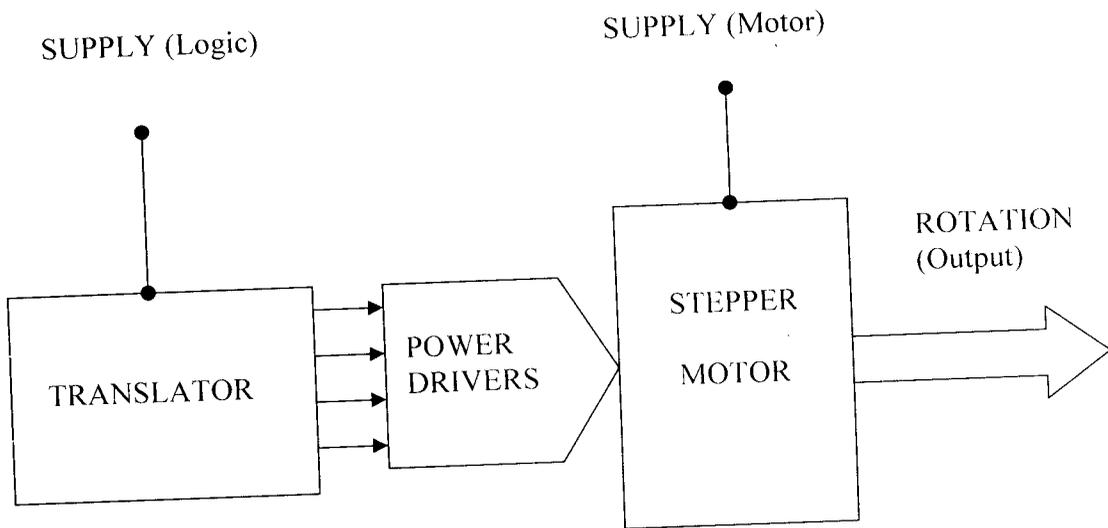


Fig 6.1 STEPPER MOTOR DRIVE CIRCUIT

A four phase stepper motor is used here. The buffers are inverting, a high on an output pin turns on current to a winding. The switching sequence to step a motor such as clockwise, as we face the motor shaft, or counter clockwise. Consider the four phases as SW1, SW2, SW3 & SW4. If SW1 and SW2 are turned on. Turning off SW1 and turning on SW3 will cause the motor to rotate one step of 1.8° clockwise. Changing to SW4 and SW3 on will cause the motor to rotate another 1.8° clockwise. Changing to SW4 and SW1 on will cause another step. After that, changing to SW2 and SW1 on again will cause another step clockwise. We can repeat the sequence until the motor has rotated as many steps clockwise as we want. To step the motor counter clockwise, work through the switch sequence in the reverse direction.

Table 6.1 EXCITATION OF STEPPER MOTOR

STEP	SW4	SW3	SW2	SW1
1	1	0	0	1
2	1	0	1	0
3	0	1	1	0
4	0	1	0	1
1	1	0	0	1

Clockwise direction ↓

Counter clockwise direction ↑

6.6 APPLICATIONS OF STEPPER MOTOR

Stepper motors can be found in almost any piece of electro-mechanical equipment. Good sources for stepper motors include:

INSTRUMENTATION

Electronic analogue watches, clocks on railway platforms/offices/factories, and camera shutter operation, photo-printing machines etc.

COMPUTER PERIPHERALS

Line printers, X-Y plotters, floppy disc and hard disc drives, paper reader and punch in a Teletype etc. Office electronics equipments like electronic typewriters, telex machines teleprinters, computer typesetting machines etc.

MACHINE TOOL CONTROLS

NC milling machines and lathes, CNC systems, drilling machines, conveyor belts etc.

HEAVY-DUTY APPLICATIONS

Indexing tables, X-Y tables, X-ray table Positioning, radiation treatment machine, CAT scanners, solar panel positioning in a satellite etc.

As stepper motor can be controlled as per the need, the project utilizes it for the controlled actions to be mimicked at the robotic arm. The arm is designed in a way that its dimension, inertia etc can be supported by the stepper motor being used.

6.7 ARM MOVEMENT

The arm movement is the actual output of the project. Depending on the angular value sent, the micro controller takes a clockwise rotation or anti-clockwise rotating decision. As the arm is connected directly to the rotor, it turns as per the requirement.

To make the arm realistic to look, project utilizes a plastic arm which is of light weight. This plastic arm is bolted with the stepper motor so that it stays intact for replicating the actions performed at the transmitter end. The material composition of the arm makes it light weight and durable.

The design of the arm is made simple in order to avoid unnecessary complications and time delay. Moreover, the stepper motor action is given higher priority than the hand design.

7. CONCLUSION

The project demonstrates a new method for controlling a robotic arm which is kept at a remote location. Normal control of arms for automations is done by using hand-held joy sticks. The controls of the joy sticks have to be mastered for controlling such robotic arm which could be unreliable of few operations. Joy sticks needs specialized cables connected to the arm for control purposes. The concept explained in the project has a clear advantage of control over a robotic arm at remote location. Also, this method of control is easier and tangible for moving type of robotic arms. The project depicts the control of robotic arm in horizontal plane which can also be extended to vertical positions, i.e., to make the remote arm do actions related to up and down motions of the controlling object. Hence, this project proposes a better method of automation control which is done in day-to-day industrial operations.



p-1998

APPENDIX 1-8051 DATA SHEET



NOVEMBER 1998

IS89C51

CMOS SINGLE CHIP 8-BIT MICROCONTROLLER with 4-Kbytes of FLASH

FEATURES

- 80C51 based architecture
- 4-Kbytes of on-chip Reprogrammable Flash Memory
- 128 x 8 RAM
- Two 16-bit Timer/Counters
- Full duplex serial channel
- Boolean processor
- Four 8-bit I/O ports, 32 I/O lines
- Memory addressing capability
 - 64K ROM and 64K RAM
- Program memory lock
 - Lock bits (3)
- Power save modes
 - Idle and power-down
- Six interrupt sources
- Most instructions execute in 0.5 μ s
- CMOS and TTL compatible
- Maximum speed: 40 MHz @ $V_{DD} = 5V$
- Industrial temperature available
- Packages available:
 - 40-pin DIP
 - 40-pin PLCC
 - 44-pin PQFP

GENERAL DESCRIPTION

The ISSI IS89C51 is a high-performance microcontroller fabricated using high-density CMOS technology. The CMOS IS89C51 is functionally compatible with the industry standard 80C51 microcontrollers.

The IS89C51 is designed with 4-Kbytes of Flash memory, 128 x 8 RAM, 22 programmable I/O lines, a serial I/O port for either multi-processor communications, I/O expansion or full duplex UART, two 16-bit timers/counters, an 8-level, two-priority-level, nested interrupt structure and an on-chip oscillator and clock circuit. The IS89C51 can be expanded using standard TTL compatible memory.

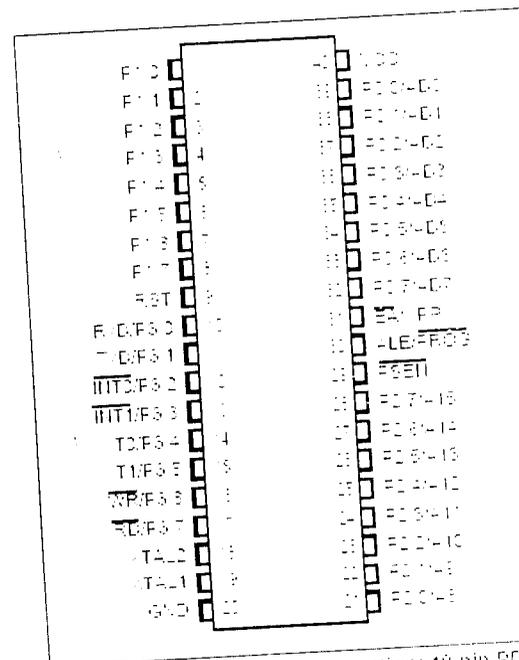


Figure 1. IS89C51 Pin Configuration: 40-pin DIP

ISSI and the ISSI logo have changed to the ISSI logo since the introduction of the ISSI logo and the ISSI logo is the only logo that can be used for the ISSI logo. The ISSI logo is the only logo that can be used for the ISSI logo. The ISSI logo is the only logo that can be used for the ISSI logo.

Table 1. Detailed Pin Description

Symbol	PDIP	PLCC	POFP	IO	Name and Function
ALE/PROG	30	32	27	IO	Address Latch Enable: Output pulse for latching the low byte of the address during an address to the external memory. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency, and can be used for external timing of clocks. Note that one ALE pulse is skipped during each access to external data memory. The pin is also the Program Pulse Input (PROG) during Flash programming.
EA/VPP	31	35	29	I	External Access enable: EA must be externally held low to enable the device to fetch code from external program memory locations 0000H to FFFFH. If EA is held high, the device executes from internal program memory unless the program counter contains an address greater than 0FFFH. This also receives the 12V programming enable voltage (VPP) during Flash programming.
PO0-PO7	39-32	43-36	37-30	IO	Port 0: Port 0 is an 8-bit open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external program and data memory. In this application, it uses strong internal pulldown when emitting 1s. Port 0 also receives the code bytes during program memory programming and outputs the code bytes during program verification. External pulldowns are required during program verification.
P10-P17	1-8	2-9	10-17 1-8	IO	Port 1: Port 1 is an 8-bit bidirectional I/O port with internal pulldown. Port 1 pins that have 1s written to them are pulled high by the internal pulldown and can be used as inputs. As inputs, Port 1 pins that are externally pulled low will source current because of the internal pulldown. (See DC Characteristics.) The Port 1 output buffers can also source four TTL inputs. Port 1 also receives the low-order address byte during Flash programming and verification.
PO2-PO7	21-28	24-31	18-25	I/O	Port 2: Port 2 is an 8-bit bidirectional I/O port with internal pulldown. Port 2 pins that have 1s written to them are pulled high by the internal pulldown and can be used as inputs. As inputs, Port 2 pins that are externally pulled low will source current because of the internal pulldown. (See DC Characteristics.) 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 to DPTR) in this application. Port 2 uses strong internal pulldown when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX to R) (n = 0-1); Port 2 emits the contents of the PO Special Function Register. Port 2 also receives the high-order bits and some control signals during Flash programming and verification. P2.6 and P2.7 are the control signals while the on-chip programs and erases.

Table 2. Special Function Registers

Symbol	Description	Direct Address	Bit Address, Symbol, or Alternative Port Function										Reset Value
ACC	Accumulator	E0-	E7	E6	E5	E4	E3	E2	E1	E0		C0-	
B	B register	F0-	F7	F6	F5	F4	F3	F2	F1	F0		C0-	
DFH	Data pointer (DFTR) high	80H										C0-	
DFL	Data pointer (DFTR) low	81H										C0-	
EA	Interrupt enable	A8-	AF	AE	AD	AC	AB	AA	A9	A8		01101000B	
			E4	—	—	E3	EA	EA	EA	EA			
PE	Interrupt priority	58-	BF	BE	BD	BC	BB	BA	B9	B8		00000002B	
			—	—	—	F5	F4	F3	F2	F1			
P0	Port 0	80H	87	86	85	84	83	82	81	80		FFH	
			P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0			
P1	Port 1	80H	97	96	95	94	93	92	91	90		FFH	
			P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0			
P2	Port 2	A0-	A7	A6	A5	A4	A3	A2	A1	A0		FFH	
			P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0			
P3	Port 3	B0-	87	86	85	84	83	82	81	80		FFH	
			P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0			
PCON	Power control	87H	SMOD	—	—	—	GF1	GF0	PD	IDL		00000002B	
PSW	Program status word	D0H	D7	D6	D5	D4	D3	D2	D1	D0		C0-	
SSBUF	Serial data buffer	88H	87	86	85	84	83	82	81	80		00000002B	
SCON	Serial controller	88H	9F	9E	9D	9C	9B	9A	99	98		C0-	
			SM0	SM1	SM2	REN	TB8	RB8	TI	RI			
SP	Stack pointer	81H										C7-	
TCON	Timer control	88H	8F	8E	8D	8C	8B	8A	89	88		C0-	
			TF1	TF0	TF1	TF0	E1	T1	E0	TF0			
TMOD	Timer mode	89H	GATE	CT	M1	M0	3-TE	2-TE	M1	M0		C0-	
TH0	Timer high 0	80H										C0-	
TH1	Timer high 1	80H										C0-	
TLC	Timer low 0	8A-										C0-	
TL1	Timer low 1	8B-										C0-	

Notes:

1. Denotes not accessible.

IS89C51

The detail description of each bits is as follow:

PSW:

Program Status Word, Bit Addressable.

7	6	5	4	3	2	1	0
CY	AC	FO	RS1	RS0	OV	—	P

Register Description:

CY	PSW: 7	Carry flag
AC	PSW: 6	Auxiliary carry flag
FO	PSW: 5	Flag 0 available to the user for general purpose.
RS1	PSW: 4	Register bank selector bit 1.
RS0	PSW: 3	Register bank selector bit 0.
OV	PSW: 2	Overflow flag.
—	PSW: 1	Usable as a general purpose flag
P	PSW: 0	Parity flag. Set/Clear by hardware each instruction cycle to indicate an odd/even number of 1 bits in the accumulator

Note

- The value presented by RS0 and RS1 selects the corresponding register bank.

RS1	RS0	Register Bank	Address
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

PCON:

Power Control Register, Not Bit Addressable.

7	6	5	4	3	2	1	0
SMOD	—	—	—	GF1	GF0	PD	IDL

Register Description:

SMOD	Double baud rate bit. If Timer 1 is used to generate baud rate and SMOD=1, the baud rate is doubled when the serial port is used in modes 1, 2, or 3.
—	Not implemented, reserve for future use.
—	Not implemented, reserve for future use.
—	Not implemented, reserve for future use.
GF1	General purpose flag bit
GF0	General purpose flag bit
PD	Power-down bit. Setting this bit activates power-down mode.
IDL	Idle mode bit. Setting this bit activates idle mode. If 1s are written to PD and IDL at the same time, PD takes precedence.

Note:

- User software should not write to reserved bits. These bits may be used in future products to invoke new features.

IE:

Interrupt Enable Register, Bit Addressable.

7	6	5	4	3	2	1	0
EA	—	—	ES	ET1	EX1	ET0	EX0

Register Description:

EA	E: 7	Disable all interrupts. If EA=0, no interrupt will be acknowledged. If EA=1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	E: 6	Not implemented, reserve for future use.
—	E: 5	Not implemented, reserve for future use.
ES	E: 4	Enable or disable the serial port interrupt.
ET1	E: 3	Enable or disable the Timer 1 overflow interrupt.
EX1	E: 2	Enable or disable External Interrupt 1.
ET0	E: 1	Enable or disable the Timer 0 overflow interrupt.
EX0	E: 0	Enable or disable External Interrupt 0.

Note: To use any of the interrupts in the 80C51 Family, the following three steps must be taken:

- Set the EA enable bit in the IE register to 1.
- Set the corresponding individual interrupt enable bit in the IE register to 1.
- Begin the interrupt service routine at the corresponding vector address of that interrupt; see below.

Interrupt Source	Vector Address
ED	0003H
—PC	0005H
E1	000BH
—E0	000DH
E0	000FH

- In addition, for external interrupts, pins INT0 and INT1, PS. 2 and PS. 3 must be set to 1, and depending on whether the interrupt is to be level or transition activated, bits IT0 and IT1 in the TCON register may need to be set to 0 or 1.
 - IT = 0 level activated
 - IT = 1 transition activated
- User software should not write to reserved bits. These bits may be used in future products to invoke new features.

IS89C51

IP:

Interrupt Priority Register, Bit Addressable.

7	6	5	4	3	2	1	0
—	—	—	PS	PT1	PX1	PT0	PX0

Register Description:

—	P.7	Not implemented; reserve for future use.
—	P.6	Not implemented; reserve for future use.
—	P.5	Not implemented; reserve for future use.
PS	P.4	Defines Serial Port interrupt priority level.
PT1	P.3	Defines Timer 1 interrupt priority level.
PX1	P.2	Defines External Interrupt 1 priority level.
PT0	P.1	Defines Timer 0 interrupt priority level.
PX0	P.0	Defines External Interrupt 0 priority level.

Notes:

- In order to assign higher priority to an interrupt, the corresponding bit in the IP register must be set to 1. While an interrupt service is in progress, it cannot be interrupted by a lower or same level interrupt.
- Priority within level is only to resolve simultaneous requests of the same priority level. From high to low, interrupt sources are listed below:
 ED
 TFD
 EI
 TF1
 R, or TI
- User software should not write to reserved bits. These bits may be used in future products to invoke new features.

TCON:

Timer Counter Control Register, Bit Addressable

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Register Description:

TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON.6	Timer 1 run control bit. Set/Cleared by software to turn Timer/Counter 1 ON/OFF.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR0	TCON.4	Timer 0 run control bit. Set/Cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when the External Interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/Cleared by software specifying edge/low level triggered External Interrupt.
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when the External Interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/Cleared by software specifying edge/low level triggered External Interrupt.

IS89C51

TMOD:

Timer/Counter Mode Control Register.
Not Bit Addressable.

	Timer 1				Timer 0			
	GATE	C/T	M1	M0	GATE	C/T	M1	M0
GATE	When TR bit in TCON is set and GATE=1, TIMER/COUNTER will run only while INT pin is high (hardware control). When GATE=0, TIMER/COUNTER will run only while TR=1 (software control).							
C/T	Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).							
M1	Mode selector bit 1.							
M0	Mode selector bit 0.							

Note 1:

M1	M0	Operating Mode
0	0	Mode 0 (13-bit Timer)
0	1	Mode 1 (16-bit Timer/Counter)
1	0	Mode 2 (8-bit auto-load Timer/Counter)
1	1	Mode 3 (Splits Timer 0 into TLO and TH0. TLO is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit Timer and is controlled by Timer 1 control bits)
1	1	Mode 3 (Timer/Counter 1 stopped)

SCON:

Serial Port Control Register, Bit Addressable.

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	T1	RI

Register Description:

SM0	SCON.7	Serial port mode specifier
SM1	SCON.6	Serial port mode specifier
SM2	SCON.5	Enable the multi-processor communication feature in mode 2 and 3. In mode 2 or 3, if SM2 is set, then RB will not be activated if the received 8th data bit (RB8) is 0. In mode 1, if SM2=1, the RI will not be activated if valid stop bit was not received. In mode 0, SM2 should be 0.
REN	SCON.4	Set/cleared by software to Enable/Disable reception
TB8	SCON.3	The 8th bit that will be transmitted in mode 2 and 3. Set/cleared by software
RB8	SCON.2	In modes 2 and 3, RB8 is the 8th data bit that was received. In mode 1, if SM2=0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
T1	SCON.1	Transmit interrupt flag. Set by hardware at the end of the eighth bit time in mode 1 or at the beginning of the stop bit in the other modes. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware at the end of the eighth bit time in mode 1 or halfway through the stop bit time in the other modes (see Note 1). Must be cleared by software.

Note 1:

SM0	SM1	MODE	Description	Baud Rate
0	0	0	Shift register	Fosc/12
0	1	1	8-bit UART	Variable
1	0	2	8-bit UART	Fosc/64 or Fosc/32
1	1	3	8-bit UART	Variable

Timer Setup

Tables 3 through 6 give TMOD values that can be used to set up Timers in different modes.

It assumes that only one timer is used at a time. If Timers 0 and 1 must run simultaneously in any mode, the value in TMOD for Timer 0 must be ORed with the value shown for Timer 1 (Tables 5 and 6).

For example, if Timer 0 must run in Mode 1 (GATE external control), and Timer 1 must run in Mode 2 (COUNTER), then the value that must be loaded into TMOD is 99H (09H from Table 3 ORed with 60H from Table 6).

Moreover, it is assumed that the user is not ready at this point to turn the timers on and will do so at another point in the program by setting bit TRX in TCON into 1.

Table 3. Timer/Counter 0 Used as a Timer

Mode	Timer 0 Function	TMOD	
		Internal Control ¹	External Control ²
0	13-Bit Timer	00H	00H
1	16-Bit Timer	01H	00H
2	8-Bit Auto-Reload	02H	04H
3	Two 8-Bit Timers	03H	05H

Table 4. Timer/Counter 0 Used as a Counter

Mode	Timer 0 Function	TMOD	
		Internal Control ¹	External Control ²
0	13-Bit Timer	04H	00H
1	16-Bit Timer	05H	00H
2	8-Bit Auto-Reload	06H	06H
3	One 8-Bit Counter	07H	0FH

Notes:

- 1 The Timer is turned ON/OFF by setting/clearing bit TR0 in the software.
- 2 The Timer is turned ON/OFF by the 1-to-0 transition on P5.0, P5.2 when TR0 = 1 (hardware control).

Table 5. Timer/Counter 1 Used as a Timer

Mode	Timer 1 Function	TMOD	
		Internal Control ¹	External Control ²
0	13-Bit Timer	10H	50H
1	16-Bit Timer	10H	60H
2	8-Bit Auto-Reload	30H	40H
3	Does Not Run	30H	60H

Table 6. Timer/Counter 1 Used as a Counter

Mode	Timer 1 Function	TMOD	
		Internal Control ¹	External Control ²
0	13-Bit Timer	40H	00H
1	16-Bit Timer	50H	00H
2	8-Bit Auto-Reload	10H	60H
3	Not Available	—	—

Notes:

- 1 The Timer is turned ON/OFF by setting/clearing bit TR1 in the software.
- 2 The Timer is turned ON/OFF by the 1-to-0 transition on P5.0, P5.2 when TR1 = 1 (hardware control).

Figure 15. SCON: Serial Port Control Register

Symbol	Position	Name and Significance	MSB						LSB
			SM0	SM1	SM2	REN	TBE	RB8	
SM0	SCON.7	Serial port mode bit 0 (see table below)							
SM1	SCON.6	Serial port mode bit 1 (see table below)							
SM2	SCON.5	Enables the multi-processor communication feature. In Modes 2 and 3, in Mode 0 only, if SM2 is set to 1, then REN will not be activated if the received 8th data bit (RB8) is 0. In Mode 1, if SM2 = 1, then REN will not be activated if a valid stop bit was not received. In Mode 0, SM2 should be 0.							
REN	SCON.4	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.							
TBE	SCON.3	The 8th data bit that will be transmitted in Modes 2 and 3. Set and cleared by software.							
RB8	SCON.2	In Modes 2 and 3, the 8th data bit that was received. In Mode 0, if SM2 = 1, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.							
TI	SCON.1	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.							
RI	SCON.0	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.							

Where SM0, SM1 specify the serial port mode as follows:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift Register	fixed (f _{cpu} /12)
0	-	1	8-bit UART	variable (set by timer)
1	0	2	8-bit UART	fixed (f _{cpu} / 32 or f _{cpu} / 20)
1	-	3	8-bit UART	variable (set by timer)

APPENDIX 2-MAX 232

194-2523, Rev. 9, 2001



+5V-Powered, Multichannel RS-232 Drivers/Receivers

Features

General Description

The MAX220-MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.23/V.24 communications interfaces, particularly applications where $\pm 12V$ is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5 μ W. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

- Portable Computers
- Low-Power Modems
- Interface Translation
- Battery-Powered RS-232 Systems
- Multichip RS-232 Networks

Superior to Bipolar

- Operate from Single +5V Power Supply (+5V and +12V—MAX231/MAX239)
- Low-Power Receive Mode in Shutdown (MAX223/MAX242)
- Meet All EIA/TIA-232E and V.24 Specifications
- Multiple Drivers and Receivers
- 3-State Driver and Receiver Outputs
- Open-Line Detection (MAX243)

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX220CFE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220CDE	0°C to +70°C	Chip
MAX220EFE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering Information continued at end of data sheet.
Contact factory for dice specifications.

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps.	Nominal Cap. Value (pF)	SHDN % Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	10	Yes	No	120	Low-power, high-speed, high-level
MAX222	+5	2/2	4	10	Yes	No	200	Low-power, high-level
MAX223	+5	2/2	4	10	Yes	Yes	120	MAX223 and receiver active in shutdown
MAX223/MAX231	+5	2/2	4	10	Yes	Yes	120	4-wire, 3-wire, 2-wire
MAX225	+5	0	0	10	Yes	No	120	6 drivers with shutdown
MAX226	+5	4	4	10	Yes	No	120	Standard $\pm 12V$ or battery supplies
MAX226/MAX227	+5	4/4	4	10	Yes	No	120	Same as MAX226
MAX228	+5 & 12	2/2	2	10	No	No	120 (120)	Industry standard
MAX229	+5 & 12	2/2	2	10	No	No	120	Industry standard
MAX229/MAX231	+5 & 12	2/2	4	10	No	No	200	High-level rate, 3-wire, 2-wire
MAX232A	+5	0/0	0	10	No	No	120	No external caps.
MAX233	+5	0/0	0	10	No	No	200	No external caps.
MAX233A	+5	0/0	0	10	No	No	200	No external caps.
MAX234	+5	0/0	0	10	No	No	120	3-wire, 2-wire
MAX234/MAX231	+5	0/0	0	10	No	No	120	Shutdown, three-state
MAX235	+5	0/0	0	10	No	No	120	Components: 3N, 2N, 1N, 1N
MAX236	+5	0/0	0	10	No	No	120	Requires 12V and 12V
MAX236/MAX231	+5	0/0	0	10	No	No	120	Standard $\pm 12V$ or battery supplies
MAX237	+5	0/0	0	10	No	No	120	Single-chip, 3-wire, 2-wire, 1-wire, 1-wire, 1-wire, 1-wire
MAX237/MAX231	+5	0/0	0	10	No	No	120	1-wire, 1-wire, 1-wire, 1-wire, 1-wire, 1-wire
MAX238	+5	0/0	0	10	No	No	120	Components: 3N, 2N, 1N, 1N
MAX238/MAX231	+5 & 12	0/0	0	10	No	No	120	Components: 3N, 2N, 1N, 1N
MAX240	+5	0/0	0	10	No	Yes	120	Open-line detection, 3-wire, 2-wire
MAX241	+5	0/0	0	10	No	Yes	200	Open-line detection, 3-wire, 2-wire
MAX242	+5	0/0	0	10	No	No	120	High-level rate
MAX243	+5	0/0	0	10	No	No	120	High-level rate, 3-wire, 2-wire, 1-wire, 1-wire
MAX244	+5	0/0	0	10	No	No	120	High-level rate, 3-wire, 2-wire, 1-wire, 1-wire
MAX245	+5	0/0	0	10	No	Yes	120	High-level rate, 3-wire, 2-wire, 1-wire, 1-wire
MAX246	+5	0/0	0	10	No	Yes	120	High-level rate, 3-wire, 2-wire, 1-wire, 1-wire
MAX247	+5	0/0	0	10	No	Yes	120	High-level rate, 3-wire, 2-wire, 1-wire, 1-wire
MAX248	+5	0/0	0	10	No	Yes	120	High-level rate, 3-wire, 2-wire, 1-wire, 1-wire
MAX249	+5	0/0	0	10	No	Yes	120	High-level rate, 3-wire, 2-wire, 1-wire, 1-wire

Maxim Integrated Products 1

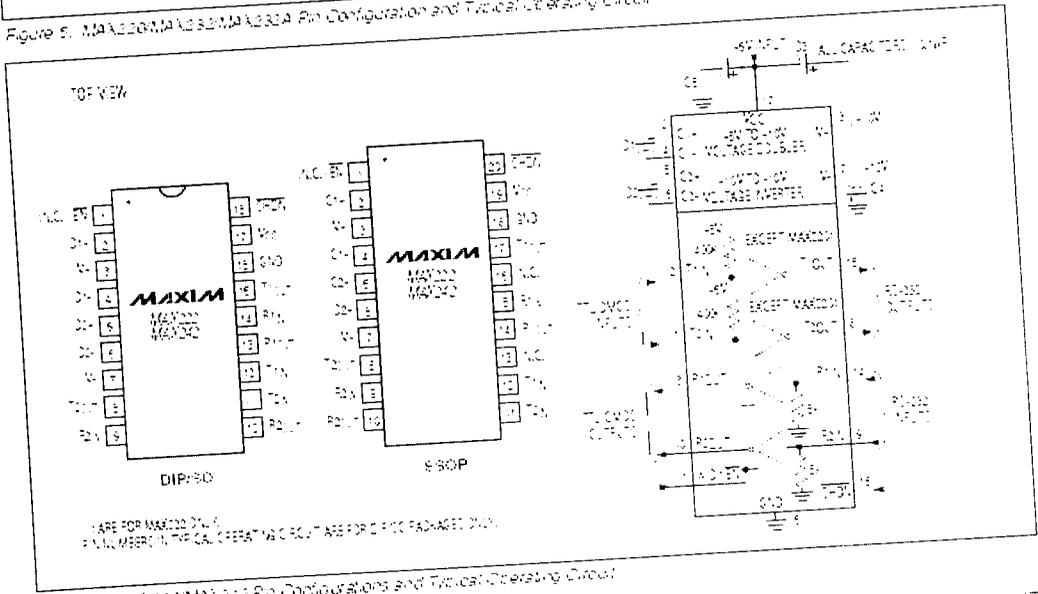
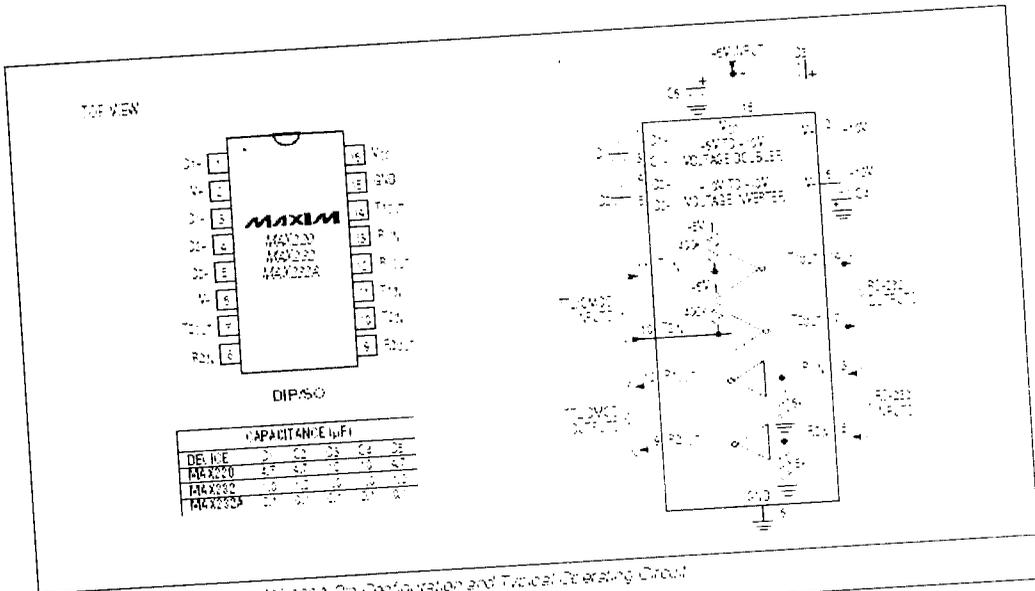


For free samples and the latest literature, visit www.maxim-ic.com or phone 1-800-998-8800.
For small orders, phone 1-800-835-8769.

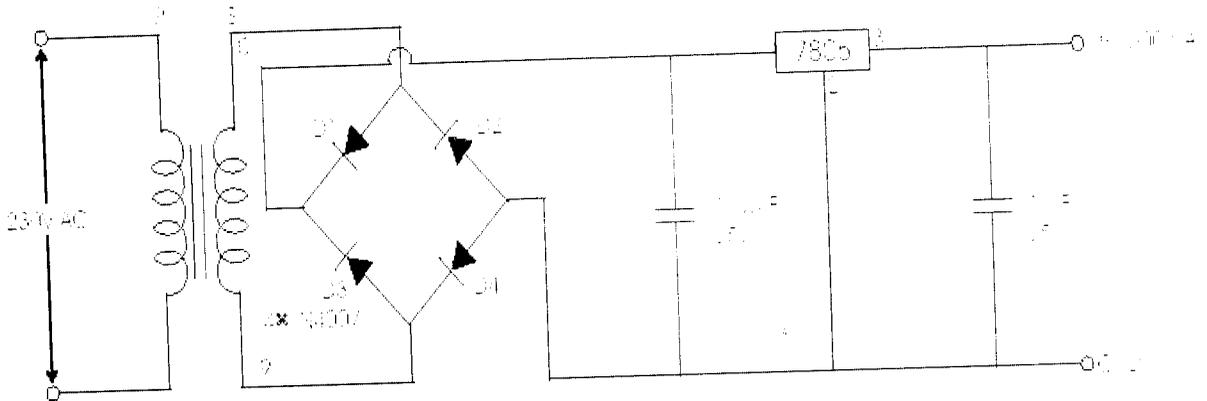
MAX220-MAX249

+5V-Powered, Multichannel RS-232 Drivers/Receivers

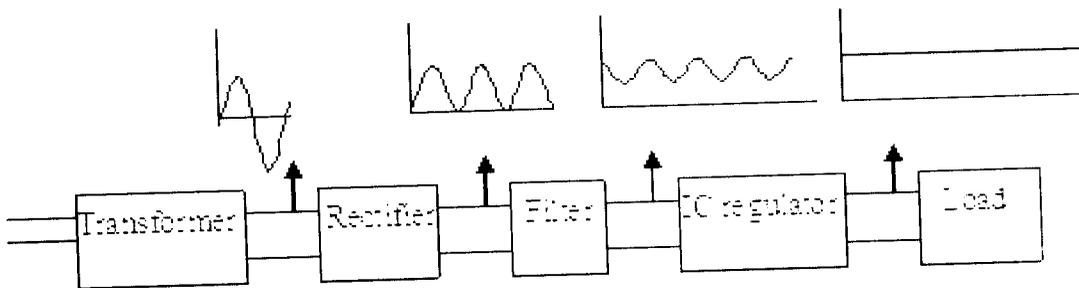
MAX220-MAX249



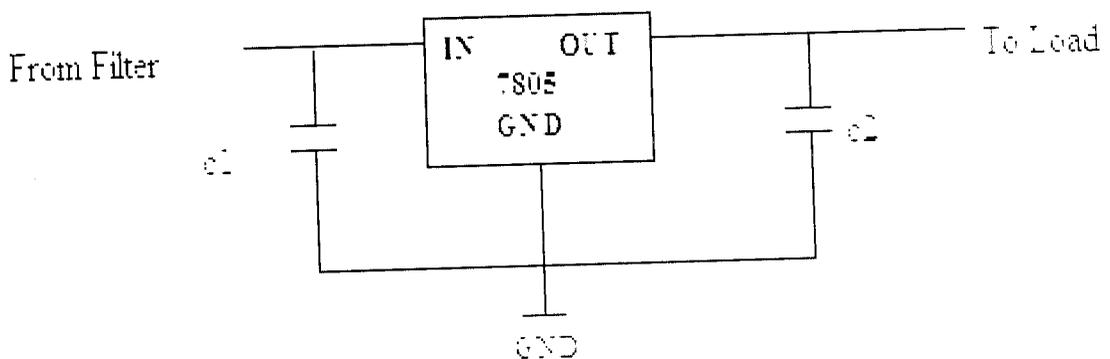
APPENDIX 3-POWER SUPPLY



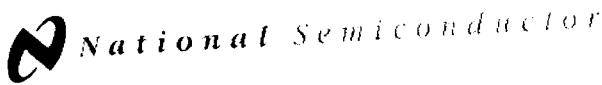
CIRCUIT DIAGRAM – REGULATED POWER SUPPLY



BLOCK DIAGRAM OF POWER SUPPLY



POSITIVE VOLTAGE REGULATOR



LM78XX Series Voltage Regulators

General Description

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expended to make the LM78XX series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the out-

put, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM78XX series provides an output voltage range from 1.2V to 37V.

Features

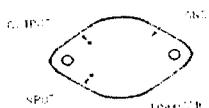
- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package

Voltage Range

LM7805	5V
LM7812	12V
LM7815	15V

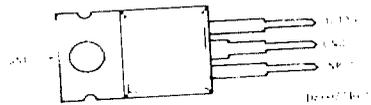
Connection Diagrams

Metal Can Package
TO-3 (K)
Aluminum



Bottom View
Order Number LM7805CK,
LM7812CK or LM7815CK
See NS Package Number KC02A

Plastic Package+
TO-220 (T)



Top View
Order Number LM7805CT,
LM7812CT or LM7815CT
See NS Package Number T03B

LM78XX Series Voltage Regulators

Absolute Maximum Ratings (Note 3)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Input Voltage (V_I) = 5V, 12V and 15V Internally Limited
 Internal Power Dissipation (Note 1) Internally Limited
 Operating Temperature Range (T_A) 0°C to +70°C

Maximum Junction Temperature 150°C
 (k Package) 150°C
 (T Package) -55°C to +150°C
 Storage Temperature Range
 Lead Temperature (Soldering, 10 sec)
 TO-18 Package k 300°C
 TO-220 Package T 280°C

Electrical Characteristics LM78XX (Note 2)

0°C, T_A = 25°C unless otherwise noted.

Output Voltage		5V			12V			15V			Units			
Input Voltage (unless otherwise noted)		10V			19V			23V						
Symbol	Parameter	Conditions			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
V _O	Output Voltage	T _J = 25°C, I _L = 1A			4.8	5	5.2	11.5	12	12.5	14.4	15	15.6	V
		R _p = 15Ω, I _L = 1A V _{MIN} V _{IN} V _{MAX}			4.75		5.25	11.4		12.6	14.25		15.75	V
V _{OL}	Line Regulation	I _L = 500 mA	T _J = 25°C	V _{IN}	0	50	4	120	4	150	-17.5	300	mV	
				0°C, T _J = +125°C	0	50	4	120	4	150	-17.5	300	mV	
		I _L = 1A	T _J = 25°C	V _{IN}	0	50	4	120	4	150	-17.5	300	mV	
			0°C, T _J = +125°C	V _{IN}	0	50	4	120	4	150	-17.5	300	mV	
V _{OL}	Load Regulation	T _J = 25°C	I _L = 1.5A 250 mA, I _L = 750 mA	V _{IN}	0	50	12	120	12	150	75	300	mV	
				5 mA, I _L = 1A, 0°C, T _J = +125°C	V _{IN}	0	50	12	120	12	150	75	300	mV
I _Q	Quiescent Current	I _L = 1A	T _J = 25°C	0°C, T _J = +125°C	0.5		8		8	8	8		mA	
				0°C, T _J = +125°C	0.5		8.5		8.5	8.5	8.5	8.5		mA
ΔI _Q	Quiescent Current Change	5 mA, I _L = 1A					1.0			1.0			1.0	mA
		T _J = 25°C, I _L = 1A V _{MIN} V _{IN} V _{MAX}			0.5		2.0	14.8		2.0	17.5		2.0	V
		I _L = 500 mA, 0°C, T _J = +125°C V _{MIN} V _{IN} V _{MAX}			0.5		2.5	14.5		3.0	17.5		3.0	V
V _n	Output Noise Voltage	T _A = 25°C, 10 Hz < f < 100 kHz					40		55			60	μV	
ΔV _{CL} ΔV _{OL}	Ripple Rejection	f = 120 Hz	I _L = 1A, T _J = 25°C 0°C, T _J = +125°C	V _{MIN} V _{IN} V _{MAX}	40		80	55		70	54		70	dB
					40		80	55		70	54		70	dB
					48		180	115		120	115		120	115
R _{OL}	Drop-out Voltage Output Resistance	T _J = 25°C, I _{DM} = 1A f = 1 MHz					20		20			19	mΩ	

Electrical Characteristics LM78XX (Note 2) (Continued)

0°C, $T_j = 125^\circ\text{C}$ unless otherwise noted.

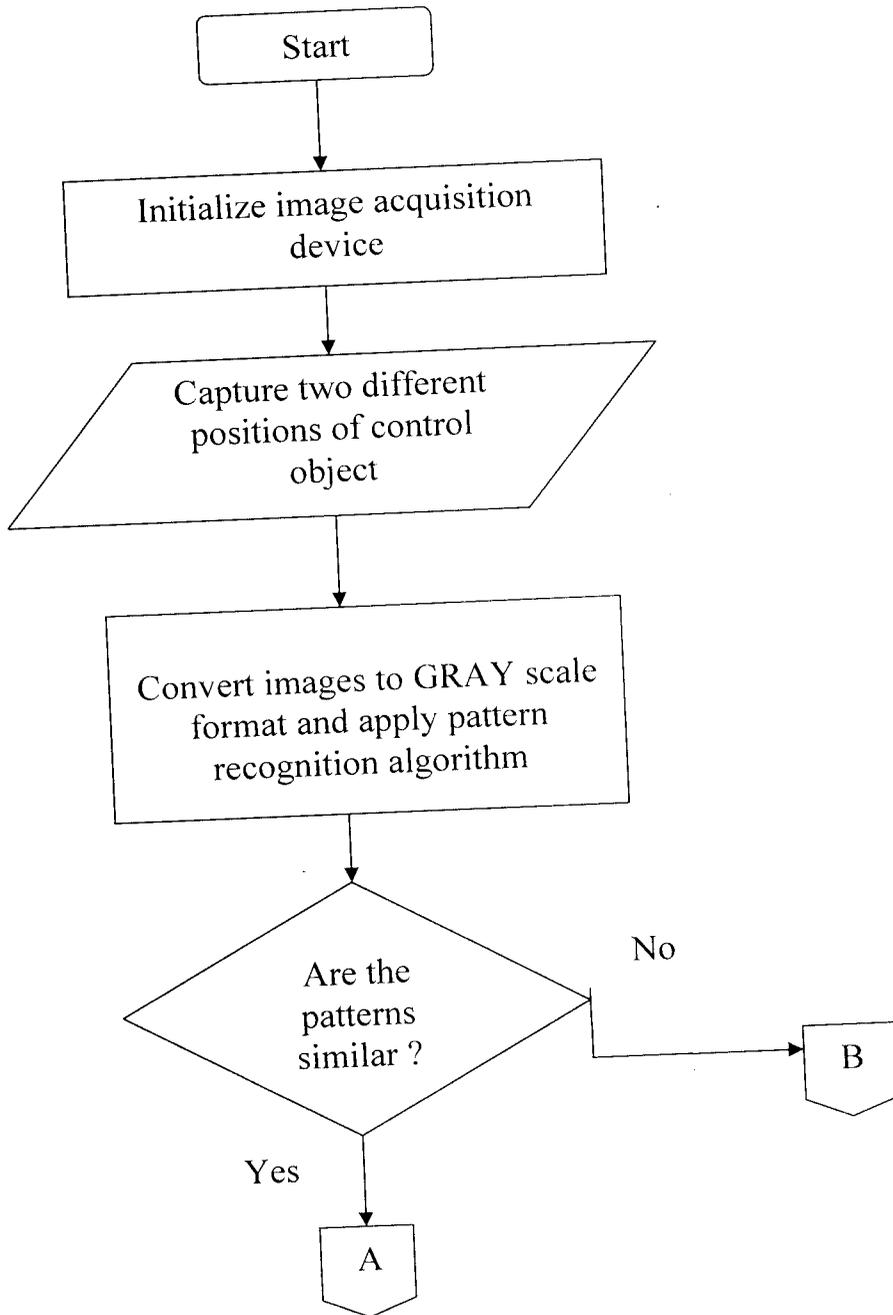
Output Voltage			5V			12V			15V			Units
Input Voltage (unless otherwise noted)			16V			19V			23V			
Symbol	Parameter	Conditions	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
	Short-Circuit Current	$T_j = 25^\circ\text{C}$	2.1			1.3			1.2			A
	Peak Output Current	$T_j = 25^\circ\text{C}$	2.4			2.4			2.4			A
	Average TC of V_{out}	0°C, $T_j = +125^\circ\text{C}$, $I_L = 5\text{ mA}$	0.6			1.3			1.8			mV/°C
Min	Input Voltage Required to Maintain Line Regulation	$T_j = 25^\circ\text{C}$, $I_L = 1\text{ A}$	7.5			14.6			17.7			V

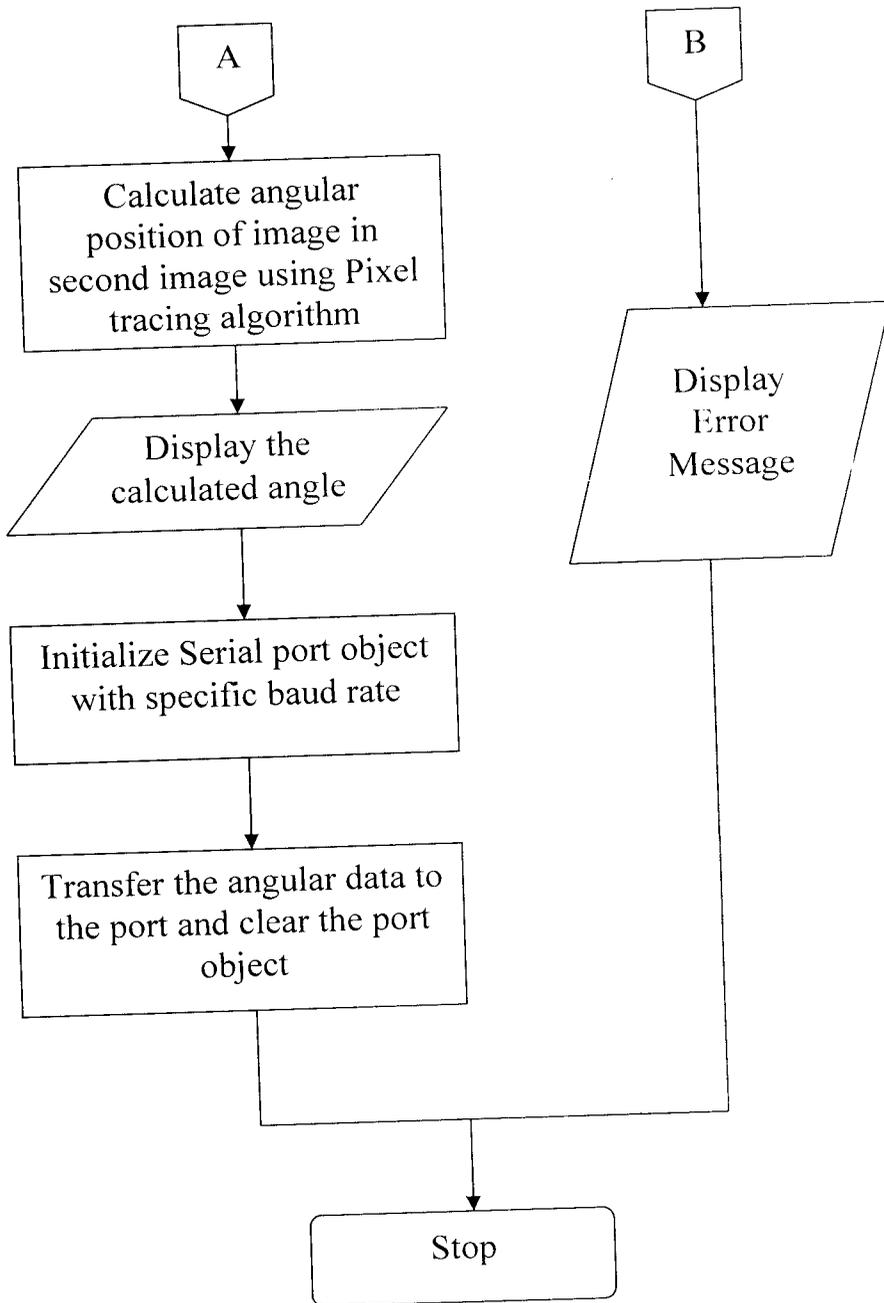
Note 1: Thermal resistance of the TO-3 package (K, K/C) is typically 4°C/W junction to case and 35°C/W case to ambient. Thermal resistance of the TO-220 package (C) is typically 4°C/W junction to case and 50°C/W case to ambient.

Note 2: All characteristics are measured with a capacitor across the input of 0.22 μF , and a capacitor across the output of 0.1 μF . All characteristics, except noise voltage and ripple rejection ratio, are measured using pulse techniques ($t_r = 10\text{ ms}$, duty cycle = 5%). Output voltage changes due to changes in internal temperature must be taken into account separately.

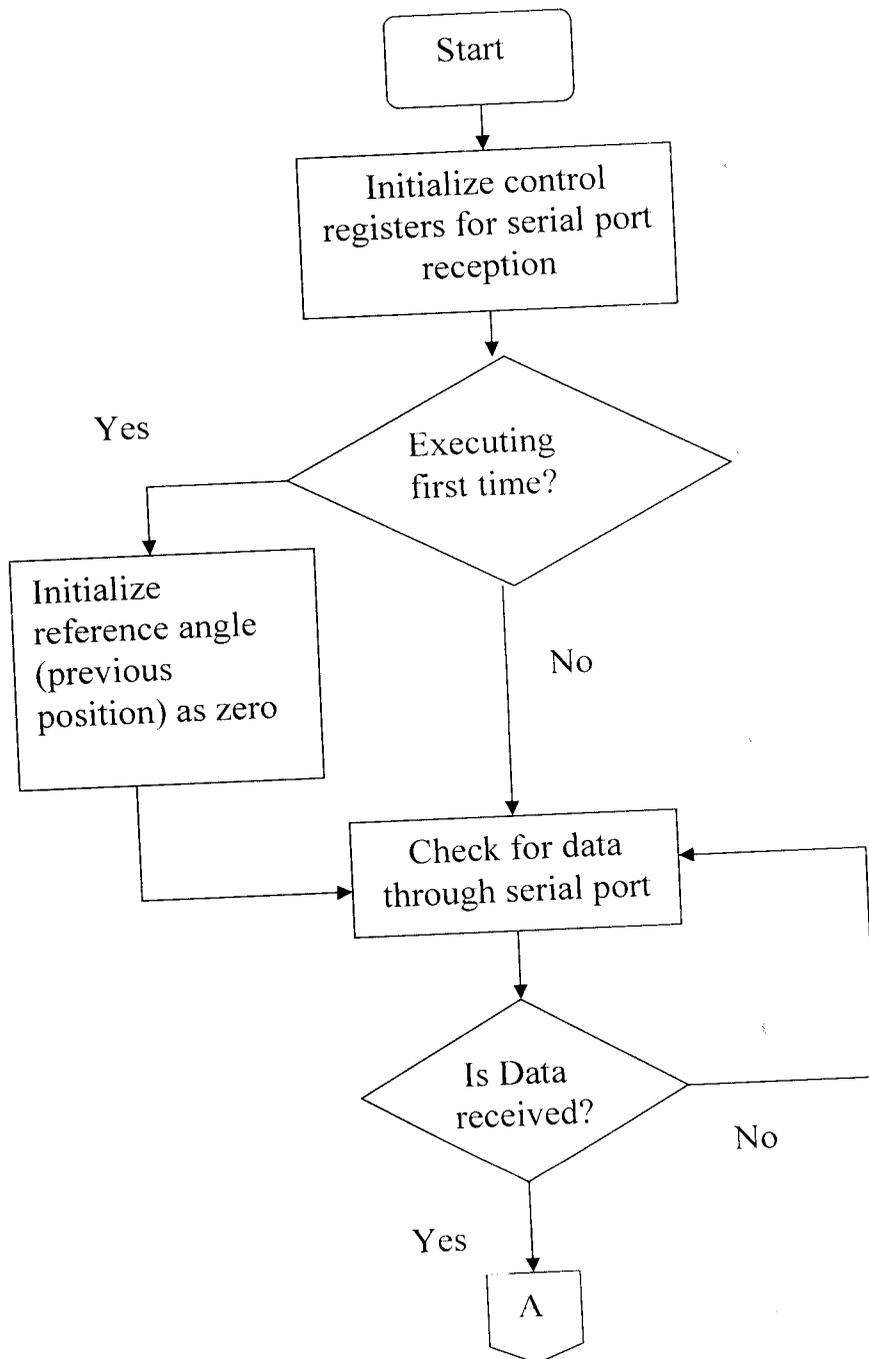
Note 3: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. For guaranteed specifications and the test conditions, see Electrical Characteristics.

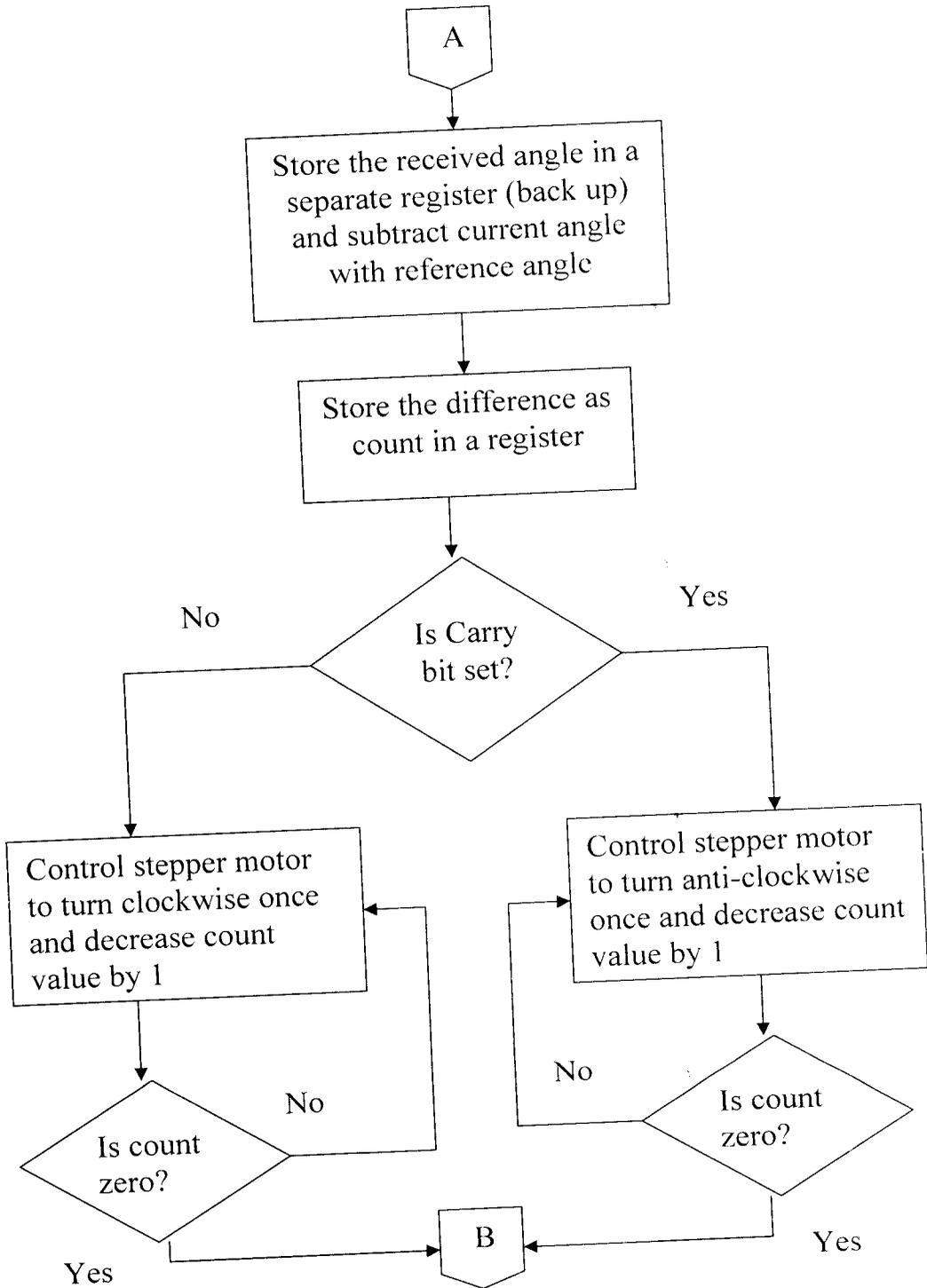
APPENDIX 4-FLOWCHART FOR MATLAB PROGRAM

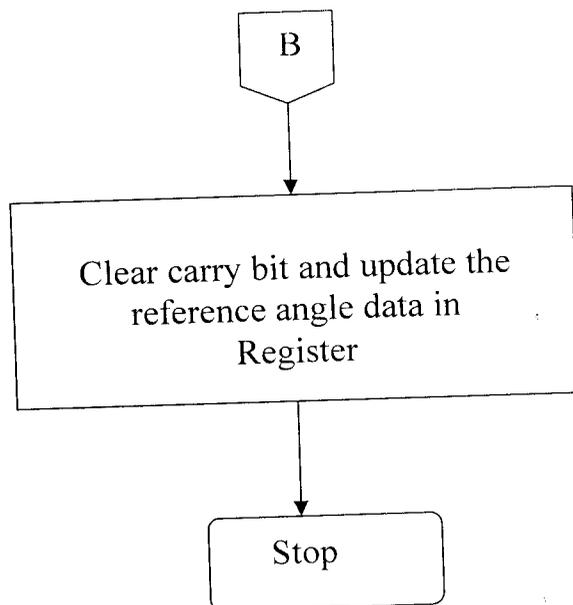




APPENDIX 5-FLOWCHART FOR MICRO CONTROLLER PROGRAM







REFERENCES

1. Rafael C. Gonzalez and Richard E. Woods (2000)
'Digital Image Processing', Addison- Wesley Publishing
Company, pp 1-15, 483-494.
2. Rafael C. Gonzalez, Richard E. Woods and Steven L.
Eddins (2004), 'Digital Image Processing Using Matlab',
Pearson Education, pp 15-75.
3. Muhammad Ali Mazidi and Janice Gillispie Mazidi
(2005), 'The 8051 Microcontroller and Embedded
Systems', Prentice Hall of India, pp 23-152, 157- 193.
4. www.mathworks.com
5. http://www.imageprocessingplace.com/DIPUM/dipum_m_files/dipum_toolbox.htm