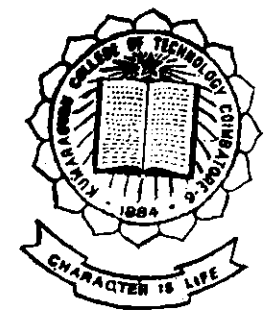
MICROCONTROLLER BASED PAGING SYSTEM

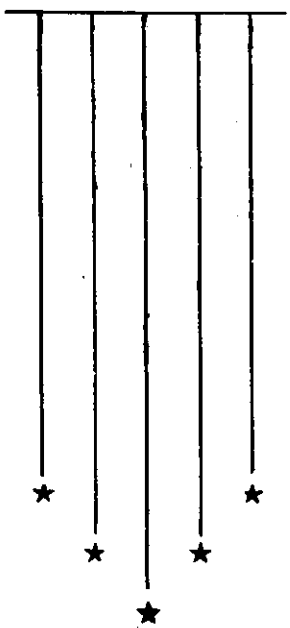


PROJECT REPORT

Submitted By

P.S.Ganesan (92 EEE 06)
Rahul Bahri (92 EEE 19)
S.Pandiyan (92 EEE 16)

Under the guidance of
Mr.V.Chandrasekaran, B.E.,



In partial fulfilment of the requirements for the award of the Degree of *BACHELOR OF ENGINEERING* in Electrical and Electronics Engineering of the Bharathiar University, Coimbatore

Department of Electrical and Electronics Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY
Coimbatore - 641 006.

1995-1996

Department of Electrical and Electronics Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore - 641 006

P-21

CERTIFICATE

S. PANDIYAN

P.S. GANESAN

Name RAHUL BAHARI

University Reg. No.....

This is to certify that the Project Work

"MICROCONTROLLER BASED PAGING SYSTEM"

is a bonafide work carried out by

Mr.....

in partial fulfilment of the requirements for the award of the **Degree of BACHELOR OF ENGINEERING IN ELECTRICAL AND ELECTRONICS ENGINEERING**, Branch of the Bharathiar University, Coimbatore during the Academic Year 1995-1996.

Station : COIMBATORE

Date : 3.4.96.

.....
Guide

.....
Head of the Department

PROFESSOR & HEAD
Dept. of Electrical & Electronics Engg.
Kumaraguru College of Tech., Coimbatore.

submitted for the University Examination held on

.....
Internal Examiner

.....
External Examiner

ACKNOWLEDGEMENT

We express our sincere thanks and profound gratitude to our Guide, **Mr.V.CHANDRASEKARAN B.E,** for his valuable guidance and immense help at all stages of this project work.

We are elated to record our heartiest indebtness to our beloved **Professor and Head, Department of Electrical and Electronics Engineering, Dr.K.A.PALANISWAMY,B.E.,M.Sc. (Engg.), Ph.D., M.I.S.T.E., C.Eng. (I), F.I.E.,** for providing us with all facilities to do this project.

We wish to express our sincere gratitude to our **PRINCIPAL, Dr.S.SUBRAMANIAN.B.E.,M.Sc. (Engg),Ph.D, S.M.IEEE.,** for extending us all possible help for the successful completion of the project.

We whole heartedly express our gratitude and sincere thanks to **Mr. PALANIAPPAN, B.E.,** of AGT Electronics and **Mr. RAMESH KUMAR, B.E.,** of United Technologies, Coimbatore.

We also wish to acknowledge our sincere thanks to our faculty members, parents and friends who have helped us in many ways throughout the course of this project.

SYNOPSIS

This project is a alpha-numeric paging system wherein a micro controller (8751) has been used.

It consists of two units namely

1. TRANSMITTER UNIT
2. RECEIVER UNIT

In the transmitting end, a computer has been utilized from which the message to be sent is converted into serial data by serial output port available in the computer. The logic level of the output is converted to digital form using IC 1489.

The serial output data is frequency modulated using frequency key shifting (FSK) technique with a carrier signal of 96.8 MHz (FM) & transmitted through a suitably designed antenna.

In the receiving end, a FM discriminator removes the carrier signal and FSK demodulator circuit gives the original message. A micro controller (8751) is used wherein a system software, written in **INTEL PL/M-51** decodes the message which starts with a receiver address followed by the message. The application software adds an end of message followed by a special character. Means are provided for scrolling & updating the messages.

This project is supported with an user friendly application software written in VISUAL BASIC in the transmitting end.

CONTENTS

CHAPTER 1

INTRODUCTION

Paging is a simple land mobile radio communication system to solve the all time problems of communication. Radio paging emerged as an useful method of communication in mid 1950's through radio communication. In 1957, the first paging system used low frequencies like 30 - 50 KHZ and its range was restricted to below 50 meters. During 1960s and 1970s on site, paging expanded rapidly with the frequency allocation of 27 - 42 KHZ bands. Both AM & FM were used which was initially limited to tone alter operation. Digital signaling became popular in the early 1970s, which was improved with numeric, alpha-numeric and voice messages in the early 1980s.

This interesting project finds a wide application in very large factories, workshops, hospitals etc. This can be used to convey messages to the concerned person who is in a remote place.The transmitter can be installed at the reception.The message can be received by the addressed person at remote place.

CHAPTER 2

2 TRANSMITTER MODULE

2.1 TRANSMITTER BLOCK DIAGRAM DESCRIPTION

The figure (2.1) shows the block diagram of a FM transmitter module. Initially the message that has to conveyed to the pager is keyed into the personal computer & broadcasted through a transmitter via a serial communication port, available in the personal computer. The choice of sending message through communication port 1 or 2 is selected by user friendly menu driven application software.

The logic level output from the personal computer is of +/- 12 Volts & that is converted into TTL logical level of 0 to 5 volts accordingly by logic level converter IC(MC1489). The precision waveform generator is used for FSK modulation. According to the logic level of FSK input, the output of waveform generator component generates two different frequencies of 1070 Hz and 1270 Hz. Any two different frequencies can be selected by the user for employing FSK technique, but the harmonics introduced may corrupt the messages. Hence to avoid the message corruption the frequencies selected are 1070 Hz and 1270 Hz which is one of the standard frequency that is used in the MODEM. The output of the waveform generator is modulated with that of FM and transmitted out into the atmosphere.

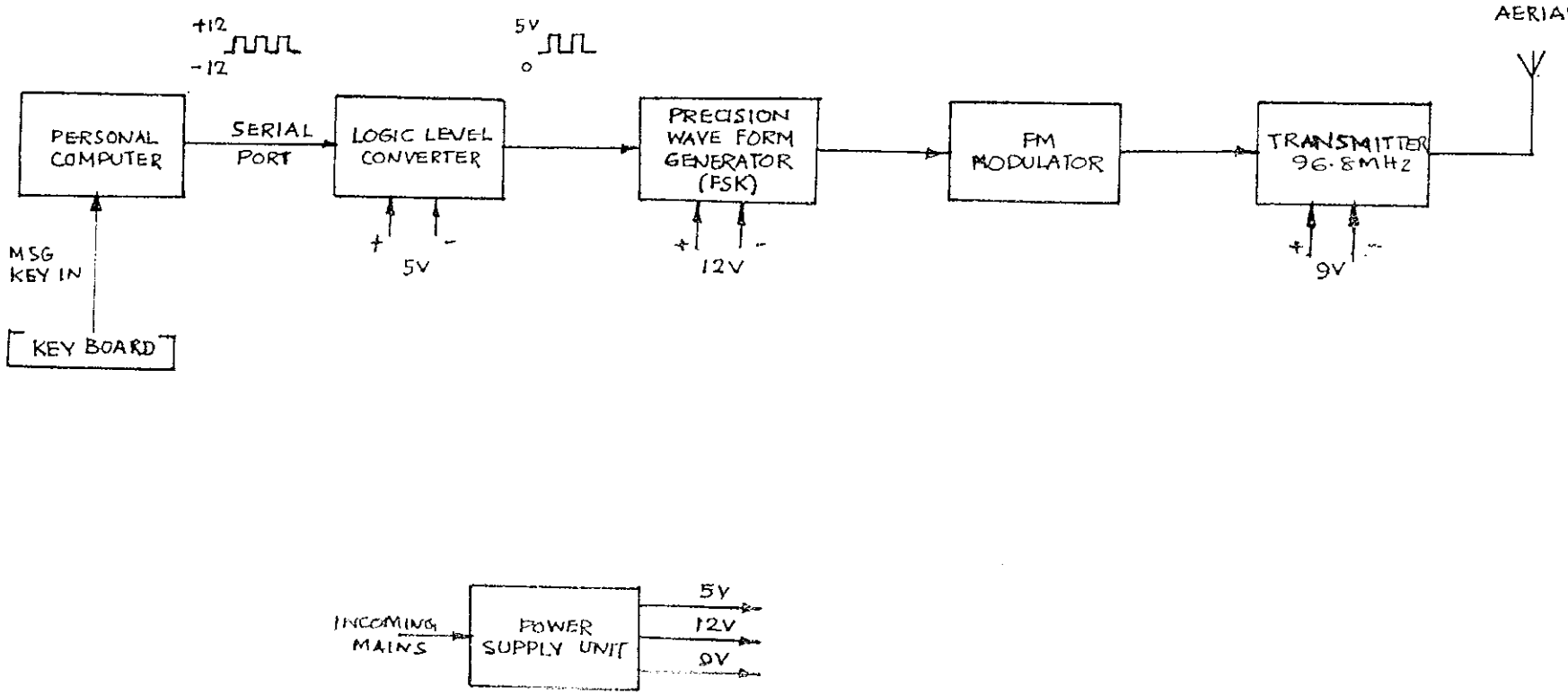


FIG: 2.1. BLOCK DIAGRAM OF TRANSMITTER.

2.2 TRANSMITTER CIRCUIT FUNCTIONAL DESCRIPTION

The transmitter module consists of the following sections.

2.2.1 POWER SUPPLY SECTION

2.2.2 LOGIC LEVEL CONVERSION SECTION

2.2.3 FSK GENERATION SECTION

2.2.4 BROADCASTING SECTION

2.1 POWER SUPPLY SECTION

The figure (2.2) shows the power supply section of transmitter. This section delivers a DC output supply of +12V, +9V and +5v output. Initially the incoming AC signal is stepped down to 18V through a step down transformer. This DC signal has also got ripples. In order to eliminate the ripples, a filtering section is incorporated.

At the regulating stage, the incoming unregulated DC supply is regulated to constant voltage of +12V, +9V & +5V by making use of IC's 7812,7809 & 7805 respectively. At this stage the output is maintained constant irrespective of any load variations or any fluctuations caused at the input of the AC mains voltage.

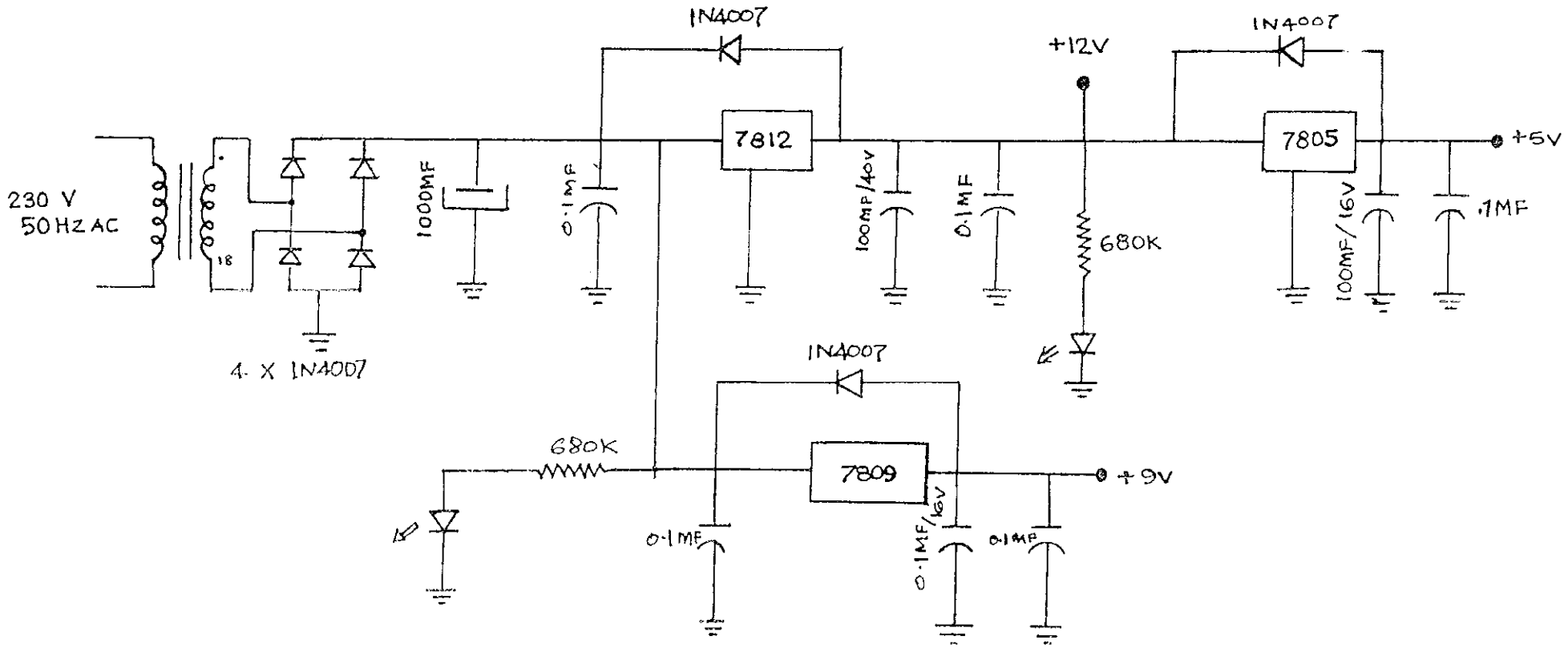


FIG 2.2 TRANSMISSION MODULE - POWER SUPPLY.

2.2.2 LOGIC LEVEL CONVERSION SECTION

When a message has to be sent to any receiver, by executing the application software, the desired message can be transmitted to the person at the remote end. When the sender requests to transmit the message, the entire telegram is made available at TXD pin of the serial port connector, which is available in the PC itself. But the threshold level of voltage would be of +/- 12 volts logic. Hence IC 1489 which is a quad line receiver is used to obtain a TTL compatible level as shown in figure (2.1). When any pulse of +/- 12volts swing is given as an input, correspondingly TTL logic level is obtained with an inversion. This level is much suitable for signal modulation.

2.2.3. FSK GENERATION STAGE

The XR 2206 IC is made used for FSK generation. This is a monolithic function generator IC capable of producing high quality sine, square, triangle and ramp waveforms of high stability and accuracy. The output waveforms can be both amplitude and frequency modulated by an external voltage.

This IC is comprised of four functional blocks, a voltage controlled oscillator (VCO), an analog multiplier and a unity gain buffer amplifier and a set of current switches.

The voltage controlled oscillator actually produces a frequency proportional to the input current, which is produced by a resistor from the timing terminals to the ground. The current switches route one of the timing pins current to the voltage controlled oscillator controlled by an FSK input pin (PIN-9), to produce an output frequency. With two timing pins (PIN 7 & 8), two discrete output frequencies can be independently produced for FSK generation.

Referring to figure (2.3.), the XR 2206 is operated with two separate timing resistors R1 & R2 connected to the timing pin 7 & 8 respectively.

Depending upon the polarity of the logic signal at pin 7, either one or the other of these timing resistors is activated. If pin 9 is open circuited or connected to bias voltage greater than or equal to 2 Volts, only R1 is activated. Similarly, if the voltage level at pin 9 is less than or equal to 1 volt, R2 is activated. Thus the output frequency can be switched between two frequencies f1 & f2.

$F1 = 1 / (R1 * C)$ &

$F2 = 1 / (R2 * C)$

The pin 3 (multiplier input) is biased midway between V+ & ground, to give output DC level of V+/2.

The potentiometers R1, pin 7 & pin 8 provides the desired frequency tuning. The maximum output swing is greater than V+/2 and the typical distortion (THD) is less than 2.5%.

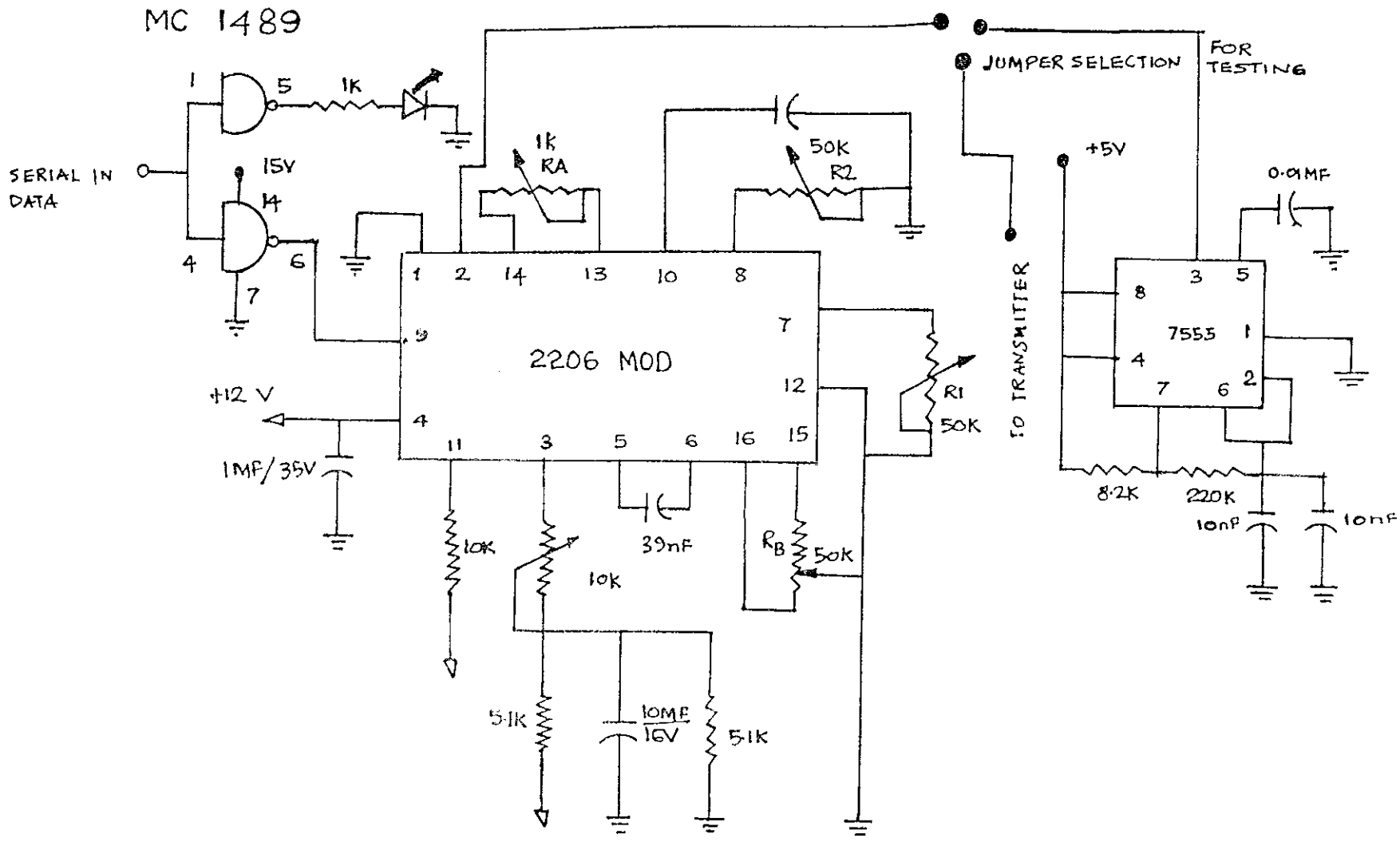


FIG 2·3 TRANSMISSION MODULE.

The harmonic content of sinusoidal output can be reduced to approximately 0.5% by additional potentiometer at the waveform adjust and symmetry adjust pins. The potentiometer Ra adjust the sine shaping resistor and Rb provides fine adjustment for the waveform symmetry.

The adjustment procedure is as follows :

1. Set Rb at mid point & adjust Ra for minimum distortion.
2. With Ra set as above, adjust Rb to further reduce distortion. Finally, the FSK generator output is obtained at pin 2 which is transmitted out into atmosphere through transmitting module.

2.2.4 BROADCASTING SECTION

The broadcasting section is shown in the figure (2.4). At this stage, the transistors & its associated components are working as common base colpitts oscillator. The FSK signal from the FSK generator circuit is directly fed to vary the capacitance of the diode D2 (Varactor diode) DB 109 through 100K resistor. Thus the capacitance of the vari cap diode is varied with FSK signal & this effect, in variation of oscillator frequency. The centre frequency of oscillator is determined by the DC voltage at the vari cap diode through 10K ohms. This voltage is varied with trimpot P1. So it is used to set the transmitting frequency. Thus according to FSK signal the frequency of transmitting signal is getting modulated.

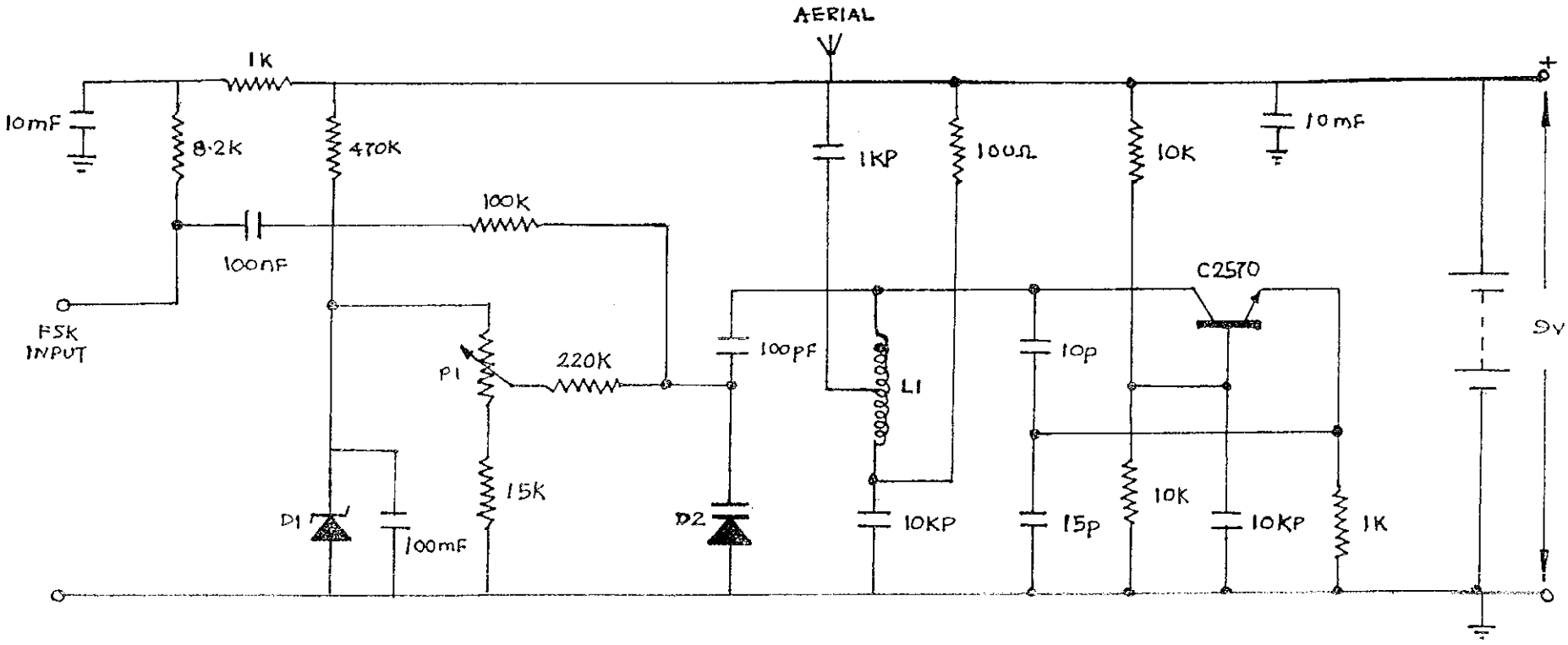
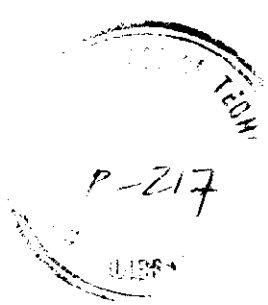


FIG 2·4 FM TRANSMITTER



RECEIVER MODULE

CHAPTER 3

RECEIVER MODULE

3.1 BLOCK DIAGRAM DESCRIPTION

Referring to the figure (3.1), the aerial circuit performs the impedance matching between the antenna and RF amplifier (FM). The signal from the aerial circuit is amplified and then applied to a mixer, via a tuned parallel circuit. The oscillator is combined with an automatic frequency control circuit (AFC), which only needs an external capacitor. The output of the mixer is applied to a two stage filter. The resulting audio frequency signal output is the FSK signal. The FSK demodulator demodulates the given FSK into corresponding stream of pulses which contains the message that is transmitted at the transmitting end. This signal is processed by micro controller and displays the message in the LCD display module with an indication.

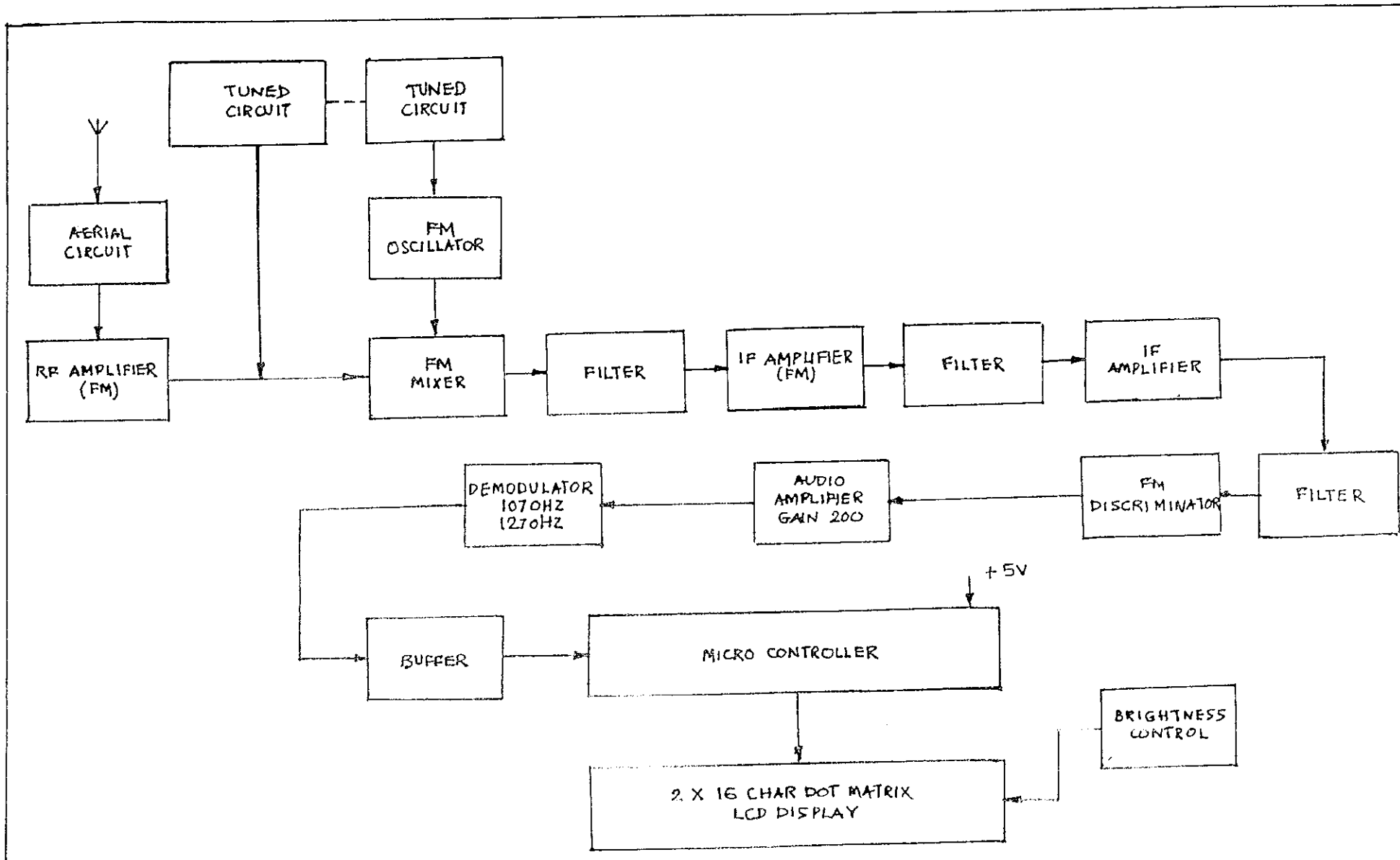


FIG 3.1 · RECEIVER MODULE.

3.2 RECEIVER CIRCUIT FUNCTIONAL DESCRIPTION :

The receiver unit comprises of following modules:-

3.2.1 FM receiver module

3.2.2 Signal amplification module

3.2.3 Message demodulation module

3.2.4 Micro controller module

3.2.5 LCD display module

3.2.1 FM receiver module

In this receiver module as in figure (3.2), it is seen that the **TEA 5591A** contains two separate receivers both superhets i.e. FM & AM. The FM section receives the incoming signal via a wide band antenna circuit & from there the signal is amplified and then applied to a mixer. The signal is mixed with oscillator signal which is also controlled by a parallel tuned circuit. The oscillator is combined with an automatic frequency control circuit (AFC) which only need an external filter and from there to the first internal amplifier (IF) and from there it is again filtered to the required level by the second internal amplifier.

Finally, the signal is demodulated in a FM discriminator. The resulting audio frequency signal is available at pin 11 of the IC. The inductor L1 (15 micro Henry) & capacitor C1 (22 pico farad) forms wide band input circuit for FM receiver. The tuned circuit for RF amplifier is formed by inductor L2 & one section of 20 pico farad of tuning oscillator.

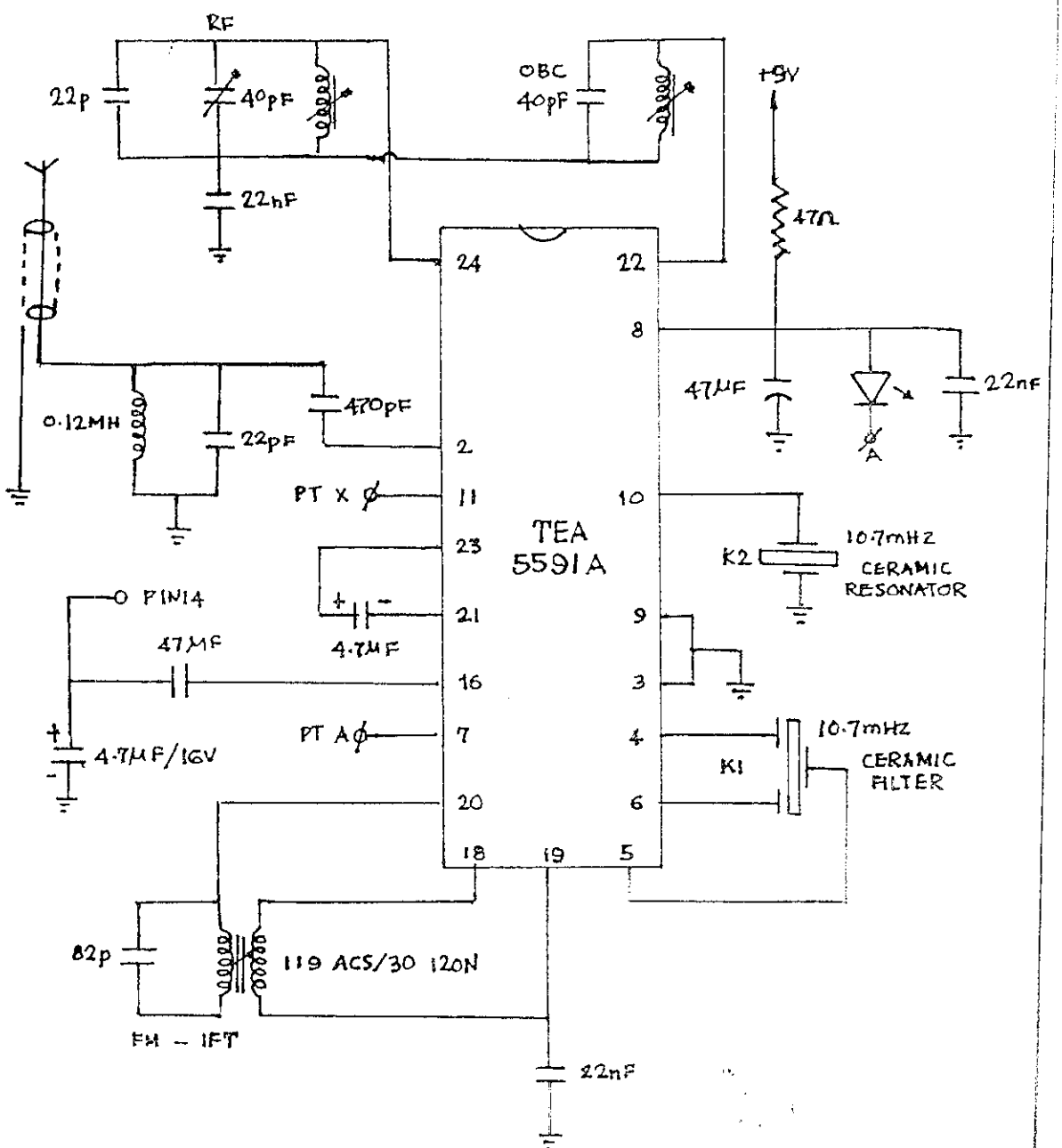


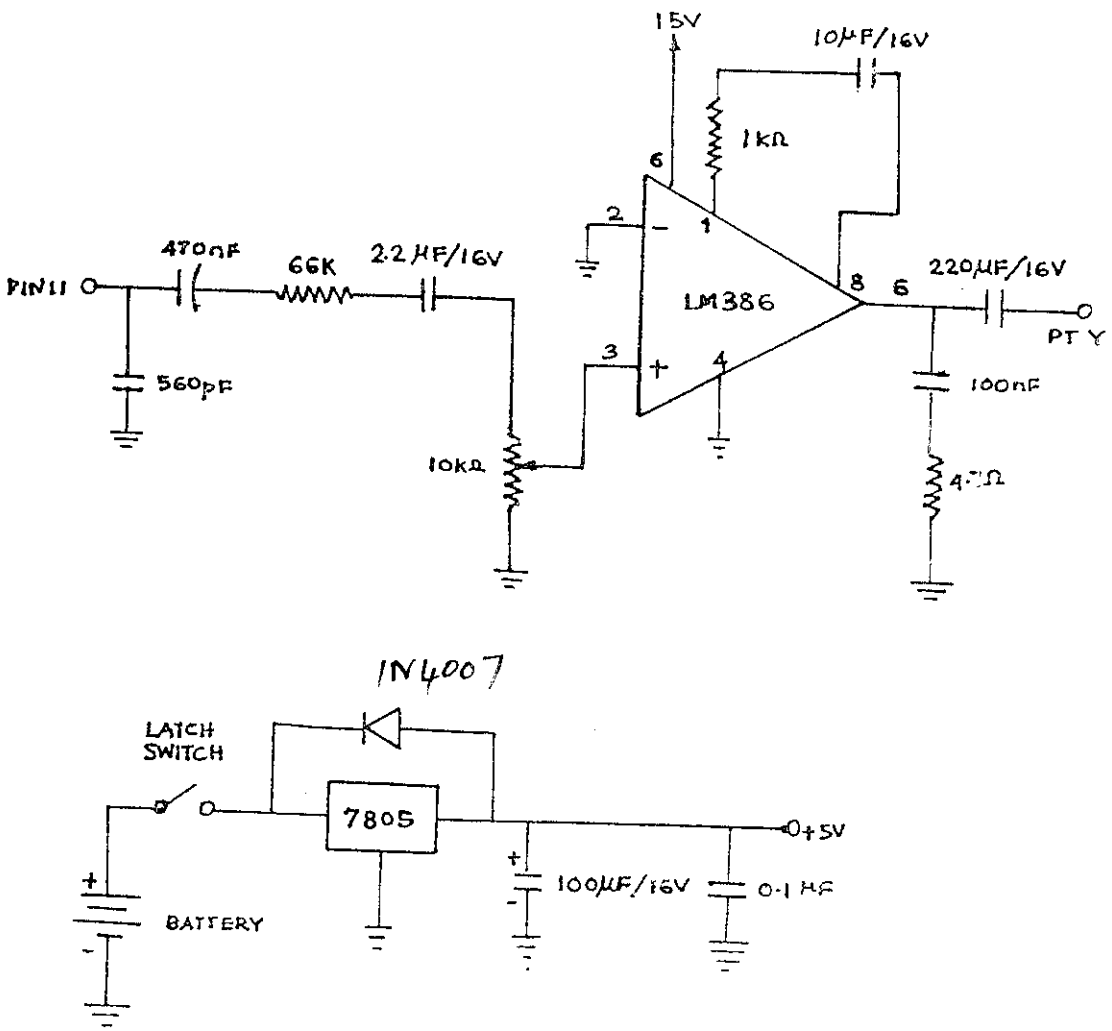
FIG 3·2 FM RECEIVER MODULE

The oscillator for the FM section is tuned by L3 & second section of tuning capacitor. The first FM - IF filter is formed by L4 & capacitor (100 pF). While the second FM - IF filter, K1 is a ceramic type. A second ceramic filter K2 ensures correct operation of FM discriminator. The 50 micro second time constant for the correct pre-emphasis is provided by internal resistance at the AF output at pin 11 (2.4K) and capacitor 680 pf.

3.2.2 SIGNAL AMPLIFICATION MODULE :

In signal amplification, a low voltage audio power amplifier is used, in which 2 pins - pin 1 & pin 8 are provided for gain control as in figure (3.3). The 1.35 Kohms internal resistor sets the gain at 2. If a capacitor is put from pin 1 to pin 8 bypassing 1.35 K resistor, the gain will go upto 200.

If a resistor is placed in series with the capacitor the gain can be set to any value from 20 to 200. The gain control is done by capacitively coupling a resistor from pin 1 to ground. For gain to be 50 we used a resistor of 1.2 K & capacitor of 10 micro farad in series with pin 1 & pin 8.



. FIG: 3.3 AMPLIFICATION MODULE.

3.3.3 MESSAGE DEMODULATION MODULE :

Message demodulation is done by the IC XR 2211. The main PLL within the IC is constructed from an input pre-amplifier, analog multiplier used as phase detector and precision voltage controlled oscillator (VCO). The pre-amplifier recognizes the input signal above typically 200 millivolts (RMS) & amplified to a constant high level signal. The multiplying type phase detector acts as a digital EX-OR gate. Its output (unfiltered) produces sum and difference frequencies of the input and the VCO output, F input + F input (2 F Input) & F and - F input to remove the "SUM" frequency component while passing the difference (DC) component to drive the VCO. The VCO is actually a current controlled oscillator with its nominal input current (FO) set by a resistor (RO) to ground and its driving current with a resistor (R1) from the phase detector.

The other sections of XR 2211 act to determine, if the VCO is driven above or below the centre frequency (FSK comparator) ; produce both active high & active low outputs to indicate, when the main PLL is in lock (quadrature phase detector & lock detector comparator). In FSK decoding, the functions of the external components are defined as follows :

Refering to the figure (3.4),

RO & CO sets the VCO centre frequency,

R1 sets the system bandwidth and

C1 sets the loop filter time constant and loop damping

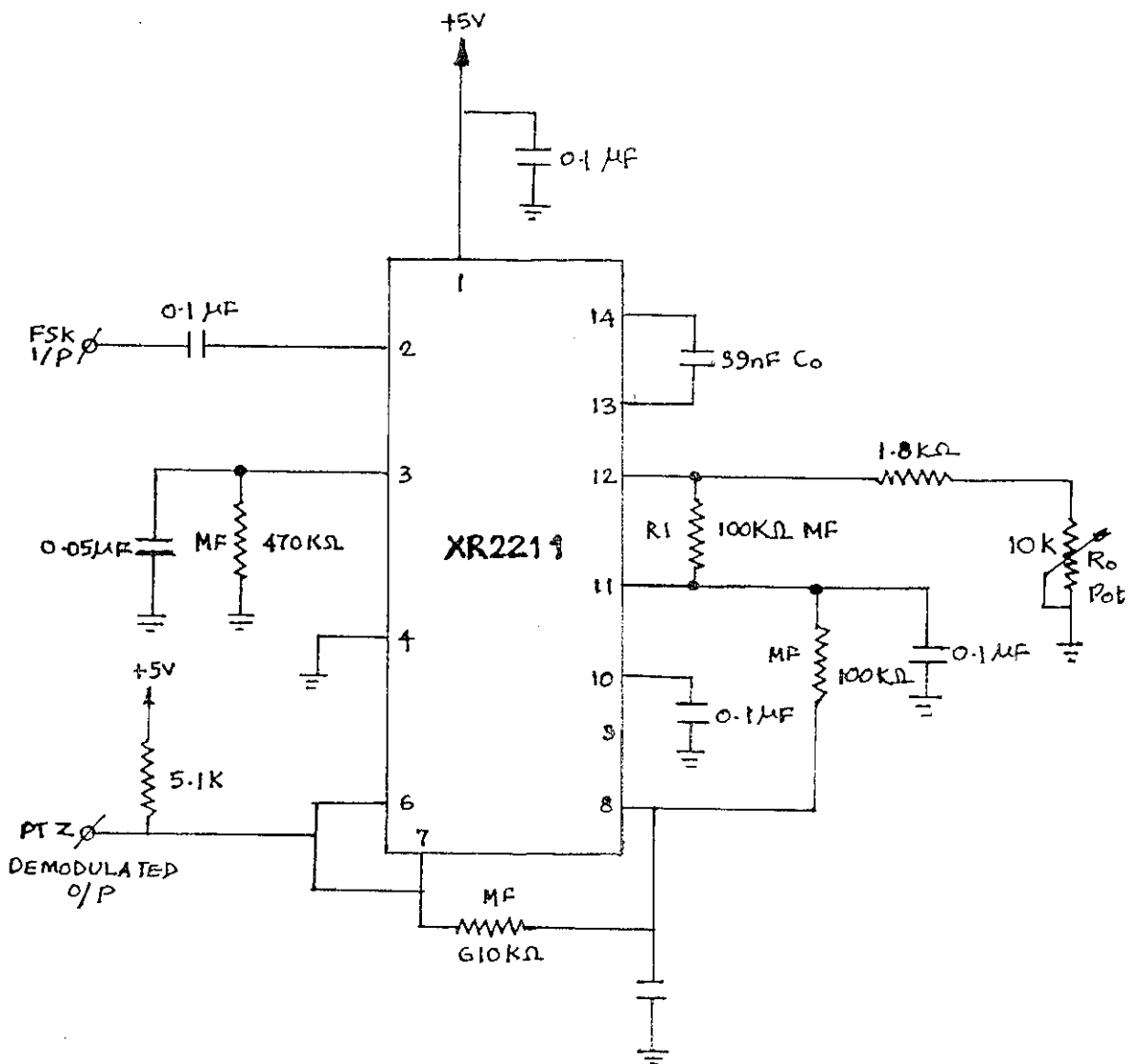


FIG:3.4 FSK DEMODULATION MODULE.

factor.

Cf & Rf form a one pole post detection filter for the FSK data output. The resistor Rb from pin 7 to 8 introduces positive feedback across the FSK comparator to facilitate rapid transition between output logic states.

3.3.4 MICRO CONTROLLER MODULE :

In the micro controller module, the software has been programmed in such a way that normally it displays a standard text on LCD screen and when any serial interrupt is received on RXD (pin 10) of the chip as indicated in figure (3.5) the interrupt is processed, so as to collect the complete message and stores into temporary memory until the "END OF THE MESSAGE" is recognized. After the completion of reception of the message, the complete telegram is analyzed for the matching of factory set address, check sum. If it matches, the message is passed over to the display RAM so that it is displayed on the LCD display & gives suitable signal to the user. The software has been supported with built in features like scrolling the messages, over to next information & also to scroll the current message. In 8751, the port 0 is used for data bus in interfacing LCD display & LSB 3 lines of port 1 i.e. P1.0 - P1.2 is used as control bus for driving the LCD controller, the port link P3.3 & P3.4 is used for sensing the two push buttons for scrolling. A resistor 8.2 K & capacitor 47 micro farad is used at reset line for proper assurance for

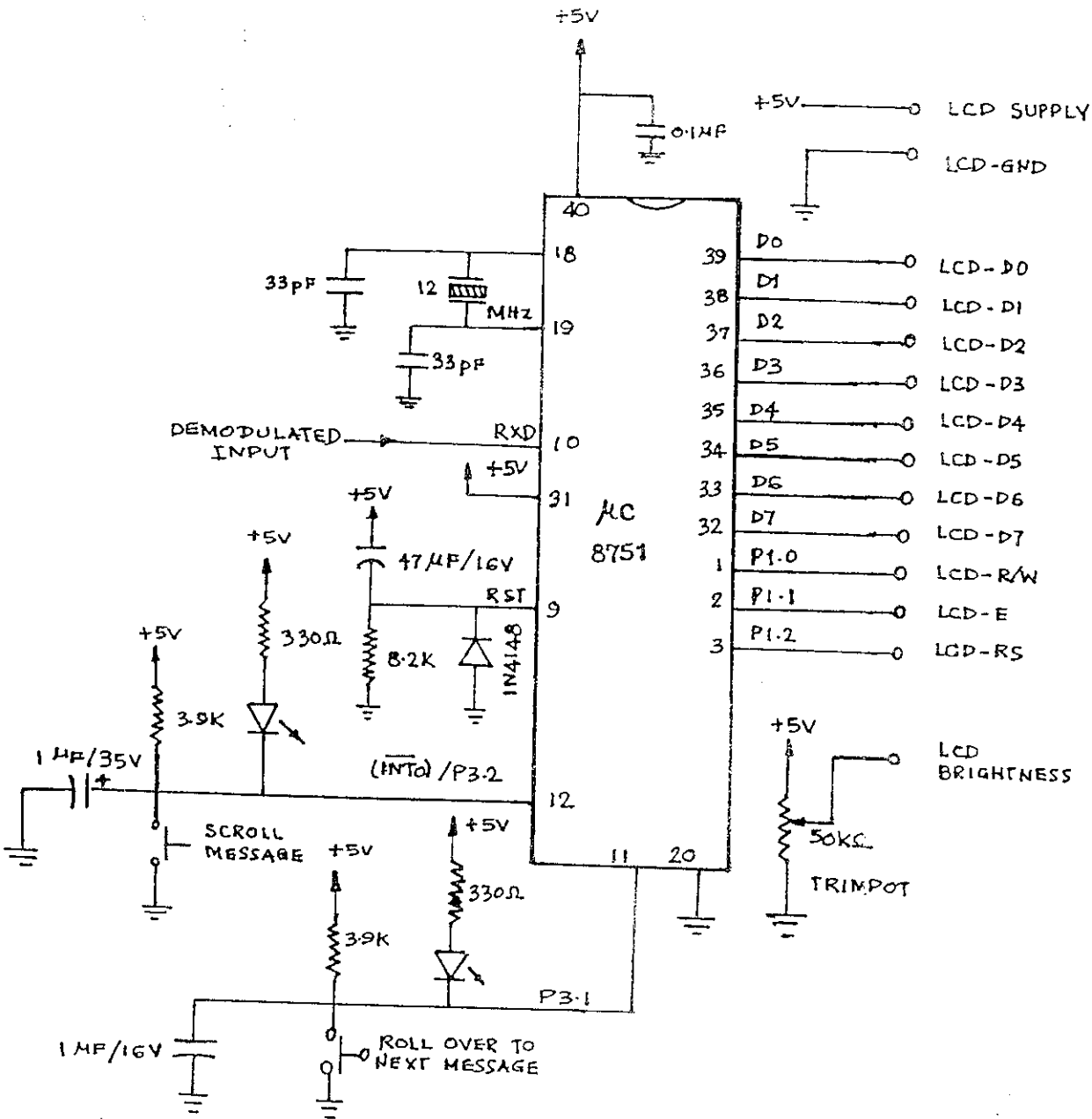


FIG 3.5. MICROCONTROLLER MODULE

the reset of micro controller.

Timer 0 is configured for 110 baud clock generation by loading "FEEBH" value in timer high and timer low byte register. In interrupt handling, the serial interrupt is given the highest priority.

3.3.5 LCD CONTROLLER MODULE :

It is a dot matrix liquid crystal display that displays alpha numeric characters & symbols. The built-in controller & driver LSI's provide convenient connectivity between dot matrix LCD & micro controller. All the functions required for dot matrix LCD drive are internally provided. Internal refresh is provided by the display module itself.

CHAPTER 4

DESIGN ASPECTS

CONCEPT

PRINCIPLE :FREQUENCY KEY SHIFTING

BAUD RATE :300BPS

LOWER FREQUENCY :1070 Hz

UPPER FREQUENCY :1270 Hz

1070 & 1270 are harmonically unrelated.

4.1 DESIGN OF TRANSMITTER MODULE

4.1.1 DESIGN OF MODULATING SECTION:

Refering figure 2.3,

Frequency of oscillation is to be determined by external timing capacitor, C, across pin 5 & 6 and by the timing resistor R connected to either pin 7 or pin 8.

LOGIC 0:

from data book, F2 = 1/ (R2*c) choosing C=0.039 MFD

Therefore R2= 1/ (1270 * 0.039 * 10 E-06)

R2=2018.97 ohms.

Hence 5 Kilo ohm trimpot is used for fine tuning.

LOGIC 1:

F1=1/(R1*C)

R1=1/(F1*C)

R1=1/(1070*0.039*10E-06)

R1=2396.35 OHMS

Hence 5 Kilo ohms trimpot is used for fine tuning.

For total distortion to be less than 2.5%, the maximum output swing is Vcc/2 . (obtained from data sheet)

ie., 12/2=6v

For THD to be less than 0.5%, the following should be adopted.

Ra = 1Kohms trimpot

Rb = 50Kohms ; is used for adjustment

Resistor Ra is sine shaping.

Resistor Rb is for fine adjustment for waveform symmetry.

ADJUSTMENT PROCEDURE :

1. Set Rb at midpoint & adjust Rb for minimum distortion

2. Set Ra as above and adjust Rb further to reduce distortion.

4.1.2 DESIGN OF FM MODULATOR:

Refering to the figure (2.4)

As it is difficult to fabricate inductors of very low value, we have used a builtin module which is a colpitts oscillator & whose carrier signal is 96.8MHz.

4.1.3 AERIAL LENGTH DETERMINATION

Length of the aerial = Quarter wavelength.

wavelength = velocity of wave/frequency

= 3 x 10 exp 8 /96.8 x 10 exp 6

= 3.099 metres

Therefore quarter wavelength = 3.099/4.

= 77.47 cm.

4.2 RECEIVER MODULE:

4.2.1 DESIGN OF MODULATING SECTION

a) Calculating PLL center frequency Fo

Fo = (F1 + F2)/2.

F1 = 1070hertz

F2 = 1270Hz

Fo = (1070 + 1270)/2

=>Fo = 1170Hz

b) as per data sheet,

Ro =20 K.ohms.

For VCO fine tuning, 18 K.ohms + 5 K.ohms trimpot can be used.

c) for finding the value Co:

VCO frequency = Fo =1/(Ro x Co);

Fo= 1170 Hz;

Ro =20 K.ohms;

Co= 1/1170 x 20 exp 3

=> 42.73 nano farads.

Co =39 nano farads

d) Calculation of R1 to give delta f equal to the mark space deviation.

From design sheet, R1= Ro x Fo / (F1 - F2)

= 20 k x 1170 / 200

R1 = 117 K.ohms.

e) To set loop damping:

Normally loop damping

t = 0.5;

Then C1 =Co /4 for t = 0.5

C1 = 0.04/4

= 0.01 ufd.

Rd is the quadrature phase detector resistor

Cd is the lock detector filter capacitance.

Cd is directly proportional to 1/capture range.

From design data, Cd = 16/capture range.

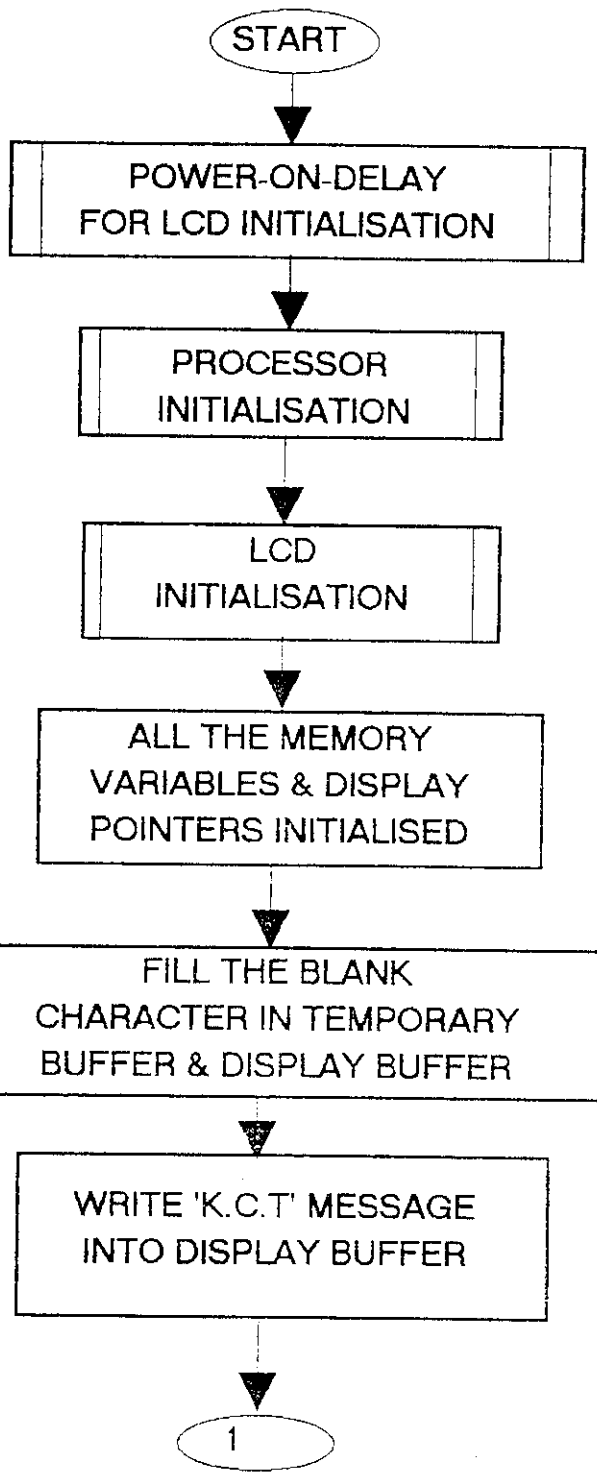
Let capture range = 250Hz

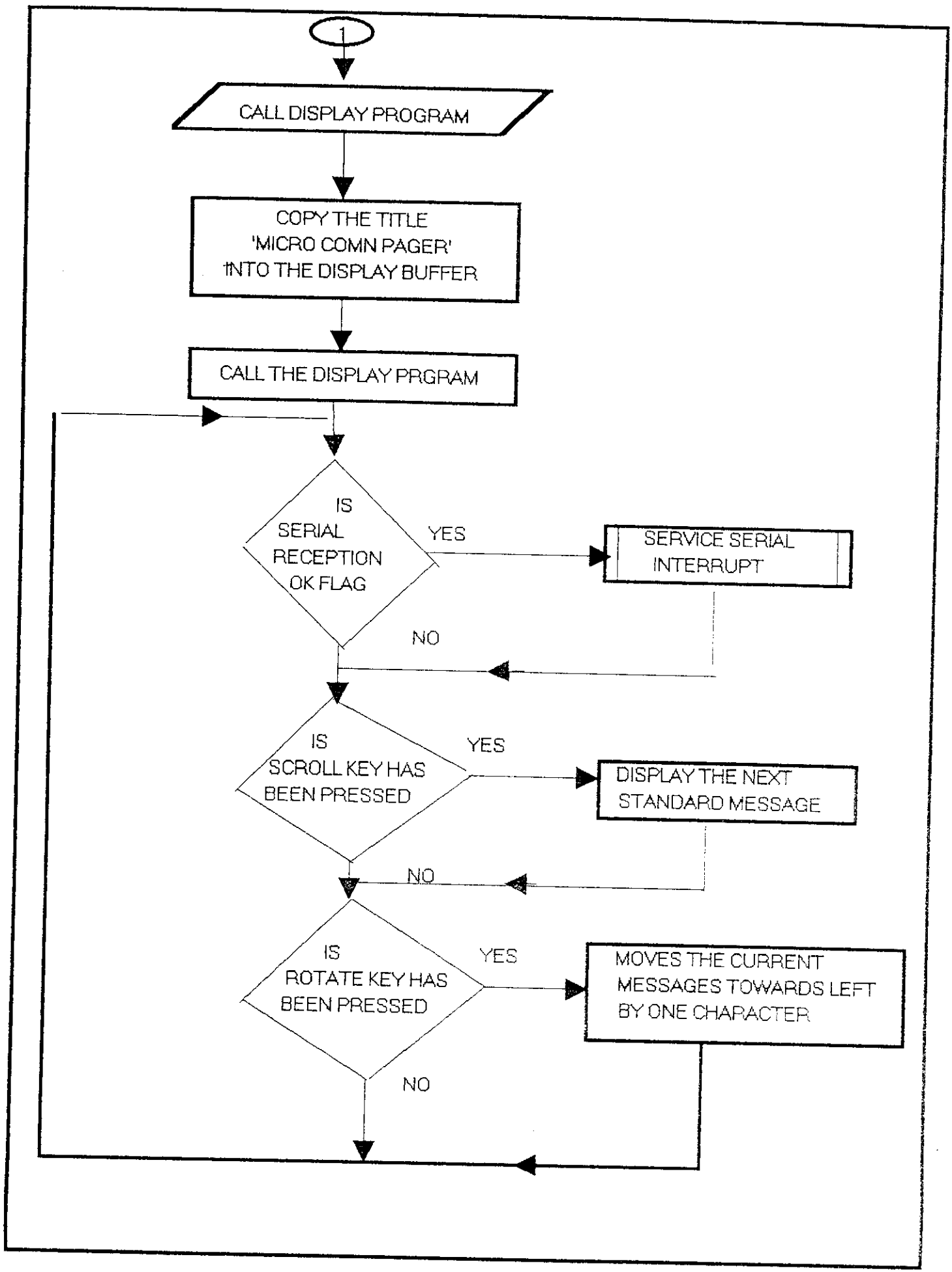
For Rd = 470 K.ohms.(STD value)

Cd = 16/250

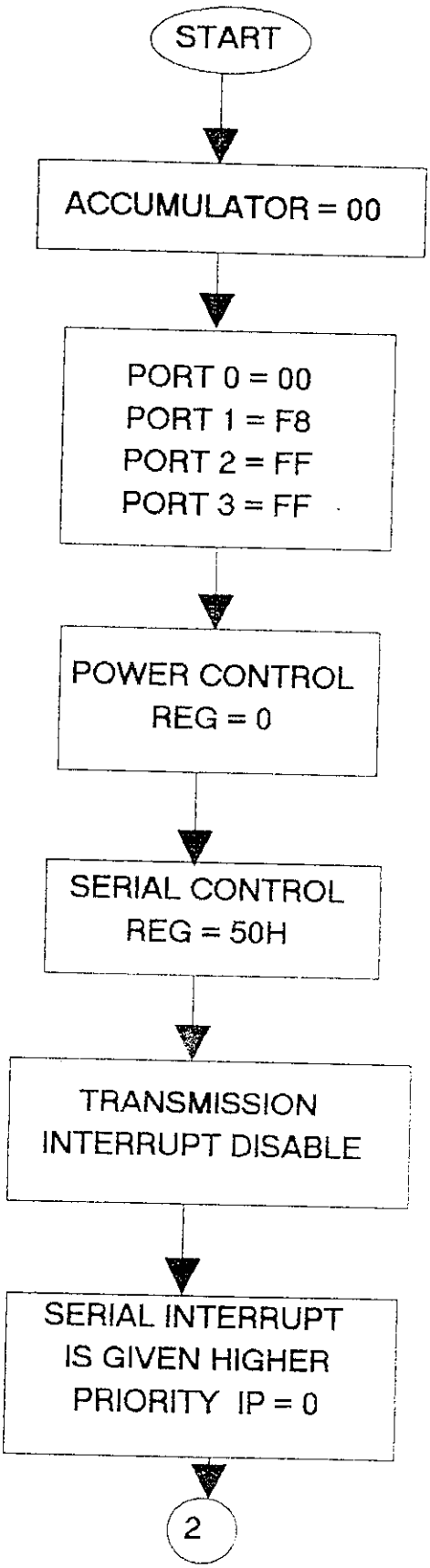
= 0.064 mfd

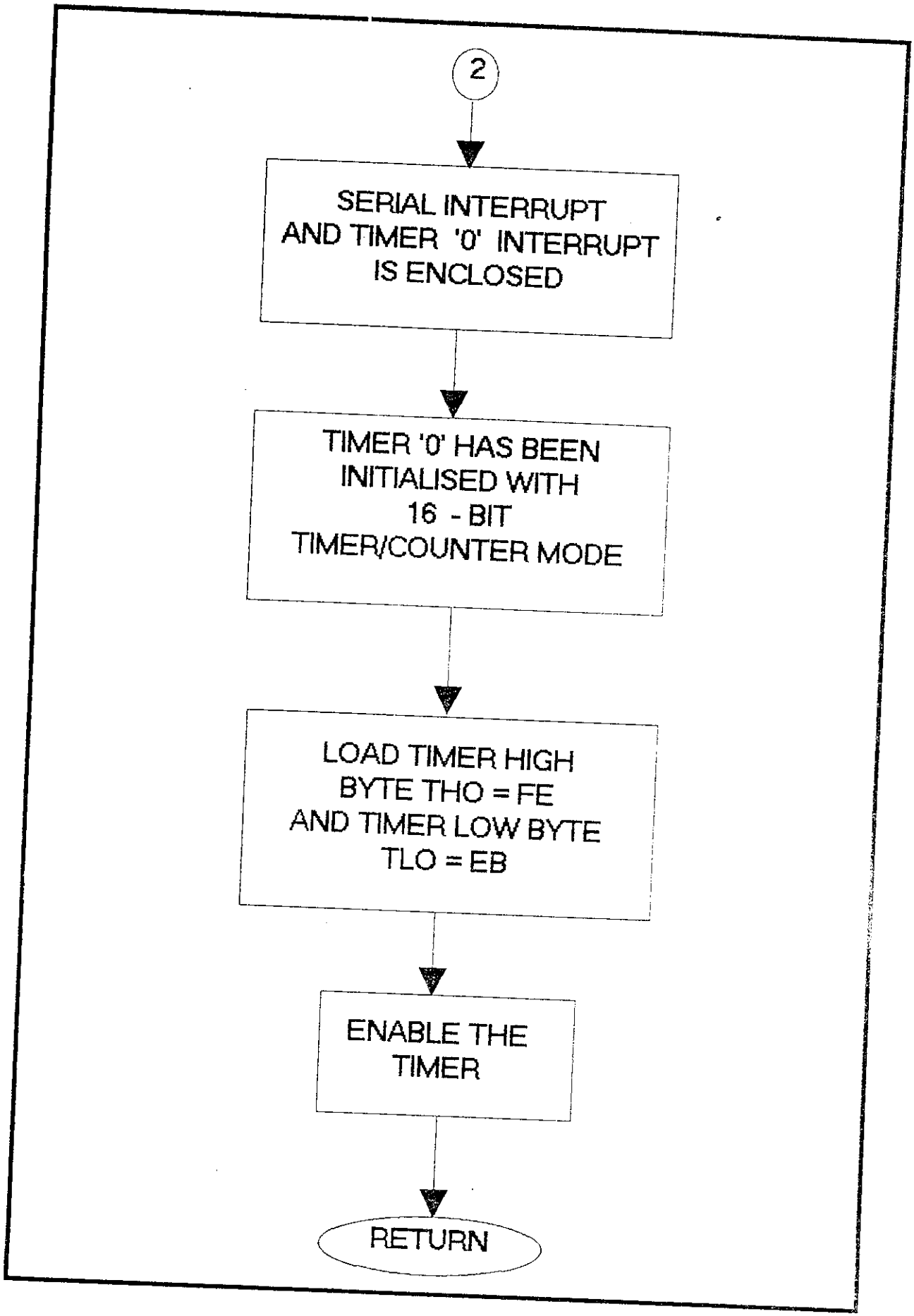
5.1 SYSTEM SOFTWARE - FLOW CHART



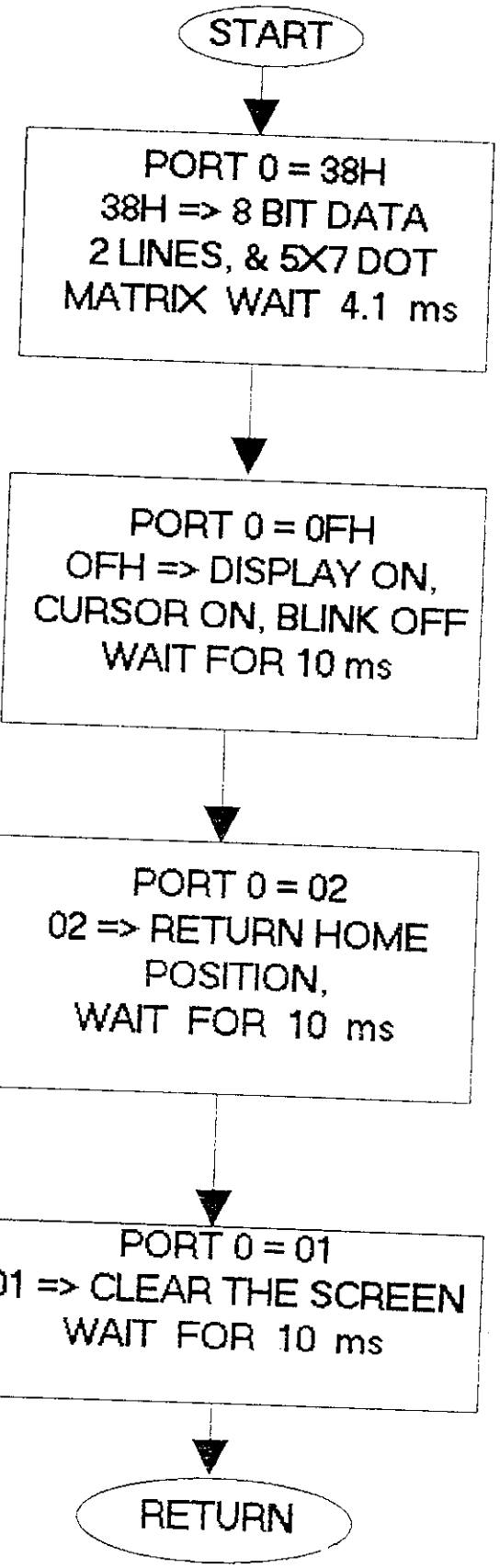

1
CALL DISPLAY PROGRAM
COPY THE TITLE
'MICRO COMN PAGER'
INTO THE DISPLAY BUFFER
CALL THE DISPLAY PRGRAM
IS
SERIAL
RECEPTION
OK FLAG
YES
SERVICE SERIAL
INTERRUPT
NO
IS
SCROLL KEY HAS
BEEN PRESSED
YES
DISPLAY THE NEXT
STANDARD MESSAGE
NO
IS
ROTATE KEY HAS
BEEN PRESSED
YES
MOVES THE CURRENT
MESSAGES TOWARDS LEFT
BY ONE CHARACTER
NO

PROCESSOR - INITIALISATION

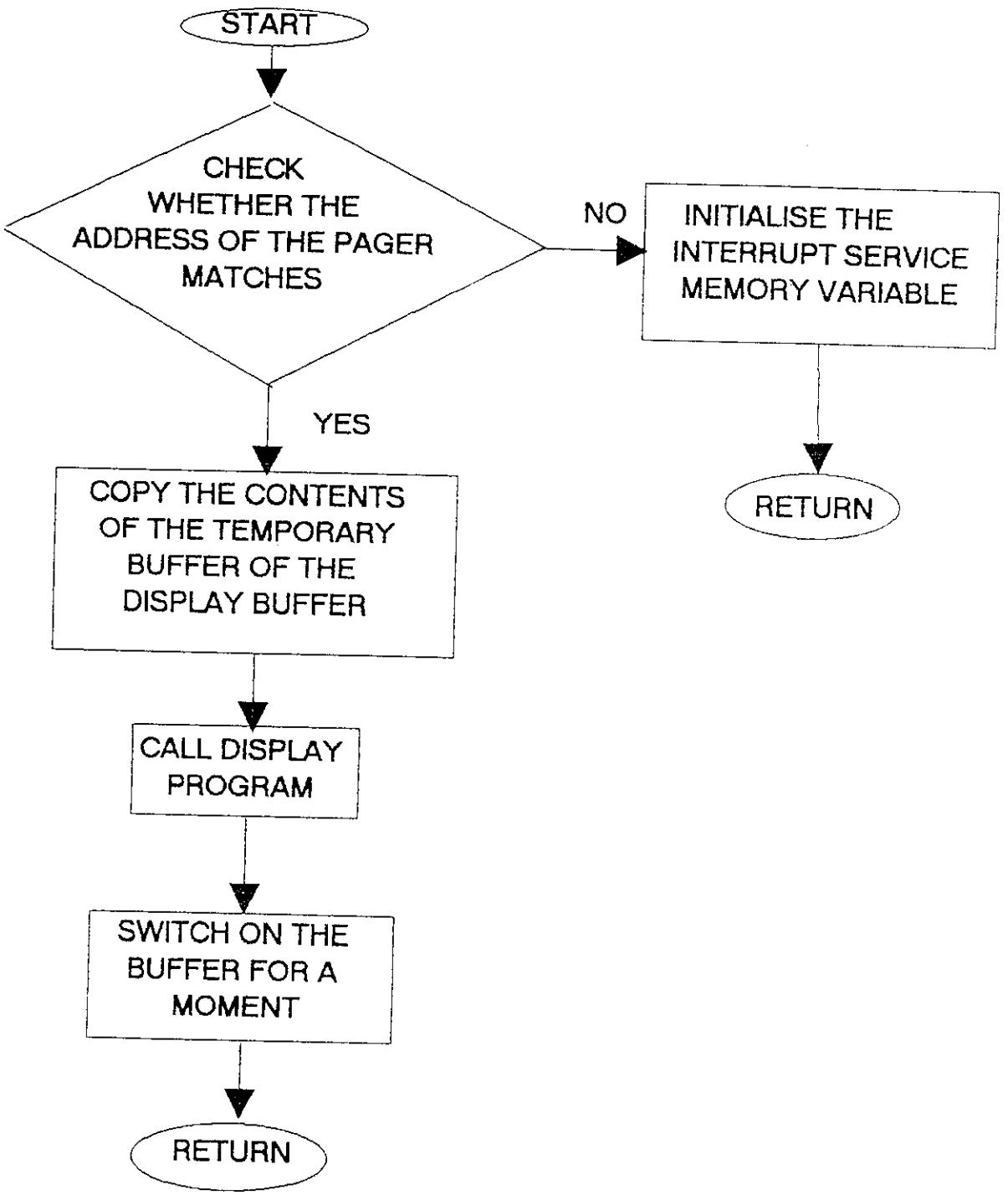



2
SERIAL INTERRUPT AND TIMER '0' INTERRUPT IS ENCLOSED
TIMER '0' HAS BEEN INITIALISED WITH 16 - BIT TIMER/COUNTER MODE
LOAD TIMER HIGH BYTE THO = FE AND TIMER LOW BYTE TLO = EB
ENABLE THE TIMER
RETURN

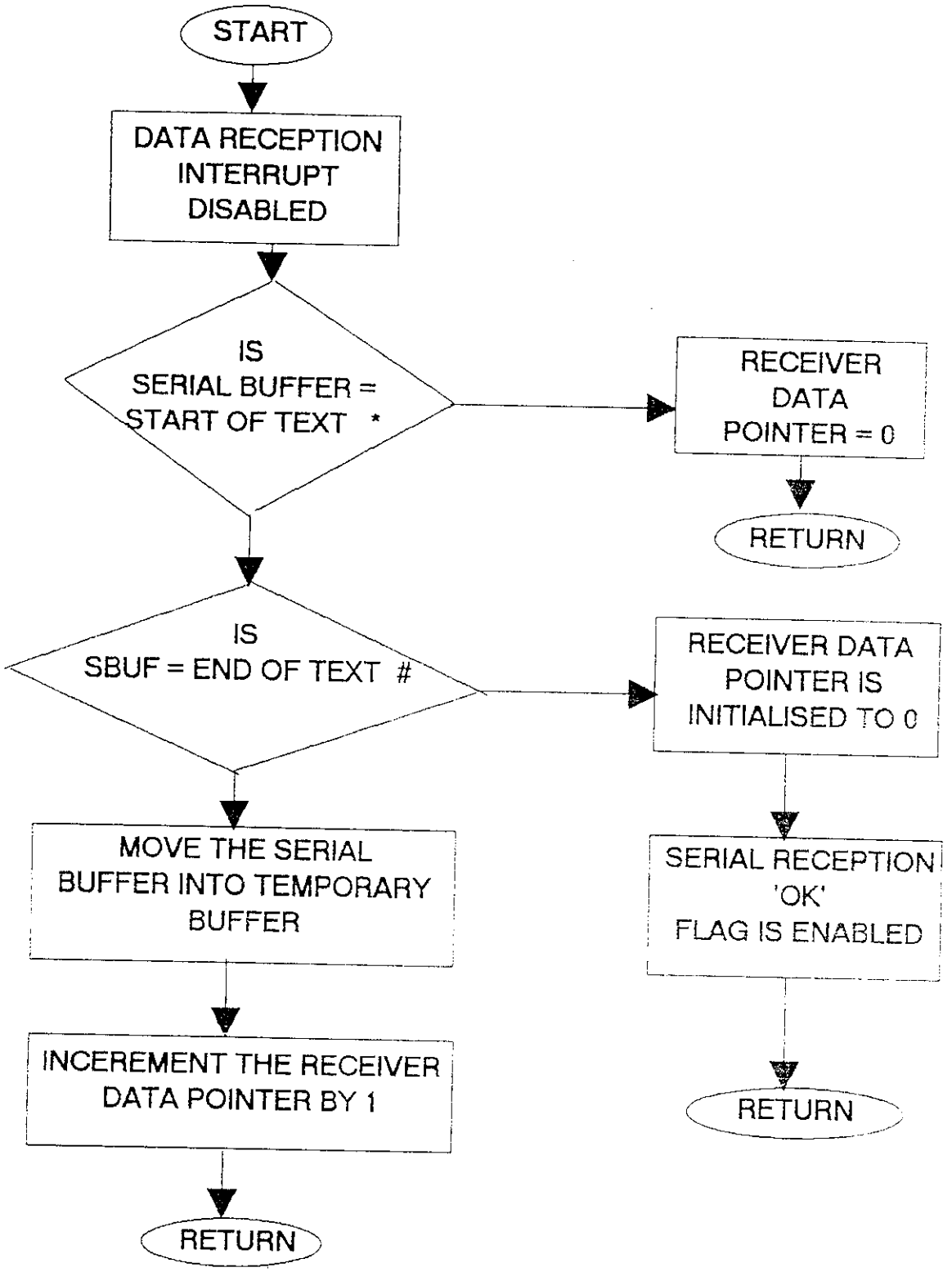
L C D - INITIALISATION



SERIAL - RECEPTION SUB - ROUTINE



SERIAL - INTERRUPT - SERVICE



```
PAGER$MAIN: DO;
$INCLUDE (C:\RAMESH\PLM51\REG52.DCL)
$CODE
      /*                                                          *
       *   PROJECT   :  MICROCONTROLLER BASED PAGER              *
       *   WRITTEN   :  BY KUMARAGURU COLLEGE STUDENTS           *
       *   STARTED   :  12/1/96                                  *
       *   FILE NAME:   KCT.SRC                                  */


DECLARE MSG1(*)  BYTE CONSTANT (' PAGER VER 1.0  ',0 );
DECLARE MSG2(*)  BYTE CONSTANT ('THANKS TO OUR HEAD OF THE
DEPT.  ',0) ;
DECLARE MSG3(*)  BYTE CONSTANT ('THANKS TO MR.V.C.S    ',0) ;
DECLARE MSG4(*)  BYTE CONSTANT ('Mr.RAHUL,MEET H.O.D SOON
',0) ;
DECLARE MSG5(*)  BYTE CONSTANT ('Mr.GANESAN MEET H.O.D SOON
',0) ;
DECLARE MSG6(*)  BYTE CONSTANT ('Mr.PANDIYAN MEET H.O.D SOON ',0) ;
DECLARE MSG7(*)  BYTE CONSTANT ('Mr.V.C.S MEET H.O.D SOON ',0) ;
DECLARE MSG8(*)  BYTE CONSTANT ('  PROJECT BY : K.C.T STUDENTS
',0) ;
DECLARE MSG9(*)  BYTE CONSTANT ('     K.C.T      ',0) ;

DECLARE TEMP$BUF(32) BYTE AT(10H);
DECLARE DISP$BUF(32) BYTE AT(30H);

DECLARE I BYTE AT (08H);
DECLARE CNTR BYTE AT (09H);
DECLARE RXDAT BYTE AT (0AH);
DECLARE GOOD BYTE AT (0BH);
DECLARE DISP$PTR WORD AT (0CH);
DECLARE EOF$BUF BASED DISP$PTR BYTE;

DECLARE SCR$KEY BIT AT (96H) REGISTER;  /* PORT P16 */
DECLARE ROT$KEY BIT AT (97H) REGISTER;  /* PORT P17 */

ADJUST$DATA: PROCEDURE (INFO$IN) BYTE;
DECLARE INFO$IN BYTE;
DECLARE INFO$OUT  BYTE;

  INFO$OUT=00H;
  IF ( INFO$IN AND 01H ) =01H THEN INFO$OUT = INFO$OUT OR 80H ;
  IF ( INFO$IN AND 02H ) =02H THEN INFO$OUT = INFO$OUT OR 40H ;
  IF ( INFO$IN AND 04H ) =04H THEN INFO$OUT = INFO$OUT OR 20H ;
```

```
   IF ( INFO$IN AND 08H ) =08H THEN INFO$OUT = INFO$OUT OR 10H ;
   IF ( INFO$IN AND 10H ) =10H THEN INFO$OUT = INFO$OUT OR 08H ;
   IF ( INFO$IN AND 20H ) =20H THEN INFO$OUT = INFO$OUT OR 04H ;
   IF ( INFO$IN AND 40H ) =40H THEN INFO$OUT = INFO$OUT OR 02H ;
   IF ( INFO$IN AND 80H ) =80H THEN INFO$OUT = INFO$OUT OR 01H ;
  RETURN (INFO$OUT);
END ADJUST$DATA;


COLLECT: PROCEDURE ;                        /* COLLECT SERIAL DATA */
   RI=0;
   IF SBUF='*' THEN                         /*  CHECK STX  */
               DO;
                    RXDAT=0;
                    END;
                ELSE
                     IF SBUF='#' THEN       /* CHECK ETX */
                                   DO;
                                   RXDAT=0;
                                   GOOD=1;
                                   END;
                                 ELSE
                                   DO;
                                   TEMP$BUF(RXDAT)=SBUF;
                                   RXDAT=RXDAT+1;
                                   END;
  IF RXDAT >=31 THEN RXDAT=31;

END COLLECT;

INTR0:PROCEDURE INTERRUPT 0;     /*  EXTERNAL INTERRUPT 0 */
END INTR0;

INTR1:PROCEDURE INTERRUPT 1;     /*  TIMER FLAG 0 */
END INTR1;

INTR2:PROCEDURE INTERRUPT 2;     /*  EXTERNAL INTERRUPT 0 */
END INTR2;

INTR3:PROCEDURE INTERRUPT 3;     /*  TIMER  FLAG  */
TH1=0FEH;
TL1=0EBH;
END INTR3;

INTR4:PROCEDURE INTERRUPT 4;     /* SERIAL INTERRUPT */
      CALL COLLECT;
END INTR4;

DELAYTIM: PROCEDURE (TIM1,TIM2);  /*  DELAY PROGRAM */
```

```
DECLARE TIM1 BYTE;
DECLARE TIM2 BYTE;
     TIM1=TIM1+1;
     DO WHILE TIM1 <>0 ;
            DO WHILE TIM2 <>0 ;
               TIM2=TIM2-1;
            END;
         TIM1=TIM1-1;
         TIM2=0FFH;
     END;
END DELAYTIM;

PWRON$DELAY: PROCEDURE ;
   DO I=0 TO 5;
    CALL DELAYTIM(200,200);
   END;
END PWRON$DELAY;

STROBE: PROCEDURE (STRBY);          /* GENERATE STROBE PULSE */
DECLARE STRBY BYTE;                  /* RS E  R/W */
                P1=P1 AND 0F8H;
          CALL DELAYTIM (10,20H);
         P1=P1 OR STRBY;
          CALL DELAYTIM (10,40H);
         P1=P1 OR STRBY OR 02H;
          CALL DELAYTIM (40,64H);
         P1=P1 AND 11111101B;
          CALL DELAYTIM (40,60H);
         P1=P1 AND 0F8H;
          CALL DELAYTIM (60,60H);
END STROBE;

CONSTR: PROCEDURE ;
       CALL STROBE(00H);
END  CONSTR;

WRDATSTR: PROCEDURE ;                  /* GENERATE WRITE PULSE */
         CALL STROBE(04H);
END WRDATSTR;

CLS: PROCEDURE ;                       /*  CLEAR SCREEN */
      P0=ADJUST$DATA(01H);
      CALL CONSTR;
      CALL PWRON$DELAY;
END CLS;

HOME: PROCEDURE ;                     /*  MOVE CURSOR TO HOME POSITION */
      P0=ADJUST$DATA(03H);
      CALL CONSTR;
```

```
      CALL PWRON$DELAY;
END HOME;

LCD$LOC: PROCEDURE (ROW,COL);     /* CURSOR LOCATE UTILITY */
DECLARE (ROW,COL) BYTE;
       IF ROW=0 THEN ROW=1;
       DO CASE ROW-1;
          ROW=0;
          ROW=40H;
          ROW=14H;
          ROW=54H;
       END;
       P0=ADJUST$DATA((ROW+COL) OR 10000000B);
       CALL CONSTR;
END LCD$LOC;

CUR$OFF: PROCEDURE ;
      P0=ADJUST$DATA(00001100B);
      CALL CONSTR;
END CUR$OFF;

DISP$CHAR: PROCEDURE (CHAR$DISP);
DECLARE CHAR$DISP BYTE;
      P0=ADJUST$DATA(CHAR$DISP);
  CALL WRDATSTR;
END DISP$CHAR;

COPY$DISP: PROCEDURE ;
DO I=0 TO 31;
   DISP$BUF(I)=TEMP$BUF(I);
END;
END COPY$DISP;

FILL$BLANK: PROCEDURE;
   DECLARE C BYTE;
   DO C=0 TO 31;
    DISP$BUF(C)='Rs';
    TEMP$BUF(C)='Rs';
   END;
END FILL$BLANK;

BLANK$TEMP: PROCEDURE;
   DECLARE C BYTE;
   DO C=0 TO 31;
    TEMP$BUF(C)='Rs';
   END;
END BLANK$TEMP;

BLANK$DISP: PROCEDURE;
```

```
   DECLARE C BYTE;
   DO C=0 TO 31;
    DISP$BUF(C)='Rs';
   END;
END BLANK$DISP;

SEC$MSG: PROCEDURE ;
         I=0;
         DO WHILE MSG9(I) <> 0;
         DISP$BUF(I)=MSG9(I);
         I=I+1;
         END;
END SEC$MSG;

COPY$TEMP: PROCEDURE (MSG$PTR);
DECLARE MSG$PTR BYTE;
CALL FILL$BLANK;
 DO CASE MSG$PTR;
      DO;
         I=0;
         DO WHILE MSG1(I) <> 0;
         TEMP$BUF(I)=MSG1(I);
         I=I+1;
         END;
      END;
      DO;
         I=0;
         DO WHILE MSG2(I) <> 0;
         TEMP$BUF(I)=MSG2(I);
         I=I+1;
      END;
      END;
      DO;
         I=0;
         DO WHILE MSG3(I) <> 0;
         TEMP$BUF(I)=MSG3(I);
         I=I+1;
         END;
      END;
      DO;
         I=0;
         DO WHILE MSG4(I) <> 0;
         TEMP$BUF(I)=MSG4(I);
         I=I+1;
         END;
      END;
      DO;
         I=0;
         DO WHILE MSG5(I) <> 0;
```

```
            TEMP$BUF(I)=MSG5(I);
            I=I+1;
            END;
        END;
        DO;
            I=0;
            DO WHILE MSG6(I) <> 0;
            TEMP$BUF(I)=MSG6(I);
            I=I+1;
            END;
        END;
        DO;
            I=0;
            DO WHILE MSG7(I) <> 0;
            TEMP$BUF(I)=MSG7(I);
            I=I+1;
            END;
        END;
        DO;
            I=0;
            DO WHILE MSG8(I) <> 0;
            TEMP$BUF(I)=MSG8(I);
            I=I+1;
            END;
        END;
  END;
END COPY$TEMP;

LCD$INIT:PROCEDURE;
    P0=ADJUST$DATA(38H);                    /* 8bit ; 2 lines ; 5*7
      dots */
    CALL CONSTR;
    CALL DELAYTIM(60,200);
    P0=ADJUST$DATA(0FH);                    /*  display on ; cursor
      on ; blink off */
    CALL CONSTR;
    CALL DELAYTIM(60,200);
    P0=ADJUST$DATA(02H);                    /* return home postion */
    CALL CONSTR;
    CALL PWRON$DELAY;
    P0=ADJUST$DATA(01H);                    /* cls */
    CALL CONSTR;
    CALL PWRON$DELAY;
END LCD$INIT;

CPU$INIT: PROCEDURE ;
         ACC=00H;
     P0=00H;
     P1=0F8H;
```

```
    P2=0FFH;
    P3=0FFH;
    PCON=0;
    SCON=50H;
    TI=0;
    IP=10H;
    IE=98H;
    EA=1;
    TMOD=10H;
    TH1=0FEH;
    TL1=0EBH;
    TR1=1;
END CPU$INIT;

MEM$INIT: PROCEDURE ;
     I=0;
     DISP$PTR=.DISP$BUF;
     CNTR=0;
     GOOD=0;
     RXDAT=0;
END MEM$INIT;

LCD$WRITE: PROCEDURE (WRITE$START);   /*  NEW LCD$WRITE */
    DECLARE WRITE$START WORD;
    DECLARE NEXT$CHAR BASED WRITE$START BYTE;
        I=0;
      DO WHILE I <> 16 ;
        IF WRITE$START >= 4FH OR NEXT$CHAR='Rs' THEN WRITE$START=30H;
        P0=ADJUST$DATA(NEXT$CHAR);
                P1=P1 AND 0F8H;
          CALL DELAYTIM (1,10H);
        P1=P1 OR 04H;
          CALL DELAYTIM (1,10H);
        P1=P1 OR 02H;
          CALL DELAYTIM (4,20H);
        P1=P1 AND 11111101B;
          CALL DELAYTIM (1,20H);
        P1=P1 AND 0F8H;
          CALL DELAYTIM (1,20H);
        WRITE$START=WRITE$START + 1;
        I=I+1;
      END;
END LCD$WRITE;

SCROLL$KEY: PROCEDURE BYTE ;        /* chk p1.6  */
DECLARE KEY$DAT BYTE;
      KEY$DAT=0;
      SCR$KEY=1;
      IF SCR$KEY=0 THEN
```

```
                        DO;
                            DO WHILE (KEY$DAT=0);
                               CALL TIME(100);
                               SCR$KEY=1;
                               IF SCR$KEY=1 THEN KEY$DAT=1;
                            END;
                        END;
      RETURN (KEY$DAT);
END SCROLL$KEY ;


ROTATE$KEY: PROCEDURE BYTE ;          /* chk p1.7  */
DECLARE KEY$DAT BYTE;
      KEY$DAT=0;
      ROT$KEY=1;
      IF ROT$KEY=0 THEN
                        DO;
                            DO WHILE (KEY$DAT=0);
                               CALL TIME(100);
                               ROT$KEY=1;
                               IF ROT$KEY=1 THEN KEY$DAT=1;
                            END;
                        END;
      RETURN (KEY$DAT);
END ROTATE$KEY ;

BUZZER$ON: PROCEDURE;                 /* chk p3.1  */
   RD=1;
END BUZZER$ON;

BUZZER$OFF: PROCEDURE;                /* chk p3.7  */
   RD=0;
END BUZZER$OFF;

CHK$SWITCH: PROCEDURE BYTE ;
DECLARE VALID$MSG BYTE;
DECLARE DISP$ADDR WORD;
DECLARE RX$NO BASED DISP$ADDR BYTE;
      DISP$ADDR=.DISP$BUF;
      VALID$MSG=0;
      IF (RX$NO='2') OR (RX$NO='9') THEN VALID$MSG=1;
      VALID$MSG=1;
      RETURN(VALID$MSG);
END CHK$SWITCH ;

      CALL PWRON$DELAY;              /* MAIN PROGRAM */
      CALL CPU$INIT;
      CALL LCD$INIT;
      CALL CUR$OFF;
```

```
        CALL PWRON$DELAY;
        CALL MEM$INIT;
        CALL FILL$BLANK;
        CALL SEC$MSG;
        CALL LCD$LOC(2,0);
        CALL LCD$WRITE(.DISP$BUF);
        CALL FILL$BLANK;
        CALL COPY$TEMP(CNTR);
        CALL COPY$DISP;
        CALL LCD$LOC(1,1);
        CALL LCD$WRITE(.DISP$BUF);

DO WHILE(1);

    IF GOOD=1 THEN
                          DO;
                             IF CHK$SWITCH=1 THEN
                                          DO;
                                          GOOD=0;
                                          CALL COPY$DISP;
                                          CALL LCD$LOC(1,0);
                                          CALL LCD$WRITE(.
                                          DISP$BUF);
                                          DISP$PTR=.DISP$BUF;
                                          CALL BLANK$TEMP;
                                          CALL BUZZER$ON;
                                          END;

                                       ELSE
                                          DO;
                                          GOOD=0;
                                          RXDAT=0;
                                          END;
                          END;


     IF SCROLL$KEY=1 THEN
                             DO;
                             IF CNTR=7 THEN CNTR=0;
                             CNTR=CNTR+1;
                             CALL BLANK$TEMP;
                             CALL BLANK$DISP;
                             CALL COPY$TEMP(CNTR);
                             CALL COPY$DISP;
                             CALL LCD$LOC(1,0);
                             CALL LCD$WRITE(.DISP$BUF);
                             DISP$PTR=.DISP$BUF;
                             END;
```

```
    IF ROTATE$KEY=1 THEN
                              DO;
                              IF ((EOF$BUF='Ps') OR (DISP$PTR >=4EH))
                                            THEN DISP$PTR=.DISP$BUF;
                                            ELSE DISP$PTR=DISP$PTR+1;
                              CALL LCD$LOC(1,0);
                              CALL LCD$WRITE(DISP$PTR);
                              CALL BUZZER$OFF;
                              END;)

 END;

END PAGER$MAIN;
```

5.3 APPLICATION SOFTWARE IN VISUAL BASIC

```
VERSION 2.00
Begin Form PG_COMM

Sub Command3D1_Click ()
Text_Comm.Text = "COM1: 110,E,8,1"
CommParameter$ = "COM1: 110,E,8,1"
PG_TXD.lbl_commset.Caption = "COM1: 110,E,8,1"
End Sub

Sub Command3D2_Click ()
Text_Comm.Text = "COM2: 110,E,8,1"
CommParameter$ = "COM2: 110,E,8,1"
PG_TXD.lbl_commset.Caption = "COM2: 110,E,8,1"
End Sub

Sub Command3D3_Click ()
Unload PG_COMM
End Sub

VERSION 2.00
Begin Form PG_MAIN

Sub client_name_Click (Index As Integer)
 PG_VNAME.Show 1

End Sub

Sub COLOR_Click (Index As Integer)

 Const CC_FULLOPEN = &H2&
  FCOLOR.Flags = CC_FULLOPEN
  FCOLOR.Action = 3
  PG_MAIN.BackColor = FCOLOR.Color
End Sub

Sub exit_Click (Index As Integer)
  PG_TXD.Comm1.PortOpen = False
  MsgBox SuperUser$ & " Thank you , Good Bye Until Next Run.
", 64, "Quit From Application"
  End    'QUIT OUT OF Application
End Sub

Sub Form_Load ()
  Cls
  CommParameter$ = "COM2: 110,N,8,2"
  SuperUser$ = " Hello Friend "
 ' Load PG_BTRFL
```

```
End Sub

Sub help_Click ()
 MsgBox SuperUser$ & " , Help not available for pager", 64,
"HELP - PAGER "
End Sub

Sub Maximum_client_Click (Index As Integer)
 PG_MAX.Show 1
End Sub

Sub name_Click (Index As Integer)
 PG_USER.Show 1
End Sub

Sub OPEN_Click (Index As Integer)

 Const OFN_READONLY = &H4&

 On Error Resume Next
   FOPEN.DefaultExt = "PAG"
   FOPEN.DialogTitle = "PAGER - File open MENU "
   FOPEN.Filter = "Text file (*.Dat) |*.Dat | All files (*.*)|
      *.* | Pager files (*.pag) |*.pag "
   FOPEN.FilterIndex = 3
   FOPEN.Flags = OFN_READONLY
   FOPEN.CancelError = False
   FOPEN.Action = 1
End Sub

Sub pager_Click ()
 PG_PAGER.Show 1
End Sub

Sub port_Click (Index As Integer)
 PG_COMM.Show 1
End Sub

Sub programmer_Click ()
 PG_PROG.Show 1
End Sub

Sub save_Click (Index As Integer)

 Const OFN_OVERWRITEPORMPT = &H4&
 Const OFN_READONLY = &H4&
 On Error Resume Next
 FSAVE.DefaultExt = "PAG"
```

```
 FSAVE.DialogTitle = "PAGER - File save MENU "
 FSAVE.Filter = "Text file (*.txt) |*.DAt | All files (*.*)|
*.* | Pager files (*.pag) |*.pag "
 FSAVE.FilterIndex = 3
 FSAVE.Flags = OFN_READONLY
 FSAVE.CancelError = False
 FSAVE.Action = 2
End Sub

Sub SPONSORSHIP_Click ()
  PG_THANK.Show 1
End Sub

Sub Stdmessage_Click (Index As Integer)
  PG_VSMSG.Show 1

End Sub

Sub Thanks_Staff_Click ()
  PG_STAFF.Show 1
End Sub

Sub transmission_Click ()
PG_TXD.Visible = True
End Sub

Sub Transmitted_Msg_Click (Index As Integer)
 'PG_VTXDMSG.Show 1
End Sub

Sub upname_Click (Index As Integer)
End Sub

VERSION 2.00
Begin Form PG_MAX

Sub SPIN_MAX_SpinDown ()

     CHKMAX = Val(LBL_MAX.Caption)
     CHKMAX = CHKMAX - 1
     If CHKMAX < 1 Then CHKMAX = 1 Else
     LBL_MAX.Caption = Str$(CHKMAX)
     MaxClientNumber = Str$(CHKMAX)

End Sub

Sub SPIN_MAX_SpinUp ()
     CHKMAX = Val(LBL_MAX.Caption)
     CHKMAX = CHKMAX + 1
```

```
     If CHKMAX > 9 Then CHKMAX = 9 Else
     LBL_MAX.Caption = Str$(CHKMAX)
     MaxClientNumber = Str$(CHKMAX)
End Sub

VERSION 2.00
Begin Form PG_PAGER

Sub BtnAboutPager_Click ()
 Unload PG_pager
End Sub

VERSION 2.00
Begin Form PG_PROG

Sub Command1_Click ()
  PG_PROG.Visible = False

End Sub

Sub Command3D1_Click ()
PG_PROG.Hide

End Sub

Begin Form PG_THANK

Sub Command1_Click ()
 PG_Thank.Hide

End Sub

Sub Command3D1_Click ()
 PG_Thank.Hide
End Sub

VERSION 2.00
Begin Form PG_TXD

Sub BtnTransCnl_Click ()
 Txmsg.Text = ""
End Sub

Sub BtnTransQuit_Click ()
PG_TXD.Visible = False
End Sub

Sub BtnTransTxd_Click ()
```

```
If Len(Txmsg.Text) > 2 And Left$(Txmsg.Text, 5) <> "ERROR" Then

  Filenumber = FreeFile
  Open "C:\RAMESH\PAGER.VB\TXDMSG.DAT" For Append As Filenumber
  Print #Filenumber, Txmsg.Text
  Close #Filenumber
  RollText = Txmsg.Text + " "
  For R = 1 To Len(RollText)
  Txmsg.Text = Mid$(RollText, R, Len(RollText) - R)
    For dly = 1 To 30000
    Next dly
  Next R
  Txmsg.Text = "" If Len(RollText) > 27 Then RollText = Mid$
    (RollText, 1, 27)
  Comm1.Output = "*" + LBL_CLIENT.Caption + RollText + "#"
  test = "*" + LBL_CLIENT.Caption + RollText + "#"
 ' Print test, Len(test)



 Else
 Txmsg.Text = "          NO MESSAGE              "
 End If
 Beep
End Sub

Sub BtnTransTxd_MouseMove (Button As Integer, Shift As Integer,
x As Single, Y As Single)
   Online.Text = "On clicking,Msg Transmitted"


End Sub

Sub Form_Load ()

    PG_TXD.lbl_commset.Caption = CommParameter$
    MaxClientNumber = Pg_Max.LBL_MAX.Caption

   ' Read all standard Messages from file

   Filenumber = FreeFile
   Open "C:\RAMESH\PAGER.VB\STDMSG.DAT" For Input As Filenumber
   While Not EOF(Filenumber)
   Line Input #Filenumber, A$
   Txmsg.AddItem A$
   Wend
   Close #Filenumber
```

```
   ' Read all user name from file

   I = 1
   Filenumber = FreeFile
   Open "C:\RAMESH\PAGER.VB\USRNAM.DAT" For Input As Filenumber
   While Not EOF(Filenumber)
   Line Input #Filenumber, CNAME$(I)
   I = I + 1
   Wend
   Close #Filenumber

 DoEvents

      '  Communication Parameter Settings

      PG_TXD.Comm1.CommPort = 2     ' com2
      PG_TXD.Comm1.Settings = "110,E,8,1"
      PG_TXD.Comm1.OutBufferCount = 0
      PG_TXD.Comm1.InputLen = 0
      PG_TXD.Comm1.PortOpen = True
End Sub

Sub LblMessage_MouseMove (Button As Integer, Shift As Integer,
x As Single, Y As Single)
 Online.Text = "Transmission Message Entry "
End Sub

Sub LblSelClient_MouseMove (Button As Integer, Shift As Integer,
x As Single, Y As Single)
   Online.Text = "Selection of client"
End Sub

Sub Spin1_SpinDown ()

 Ctnum = (Val(LBL_CLIENT.Caption) - 1)
 If (Ctnum < 0) Then Ctnum = 0 Else
            LBL_CLIENT.Caption = Str$(Ctnum):
            Text_usr.Caption = "   " & CNAME$(Ctnum)
 If Ctnum = 0 Then Text_usr.Caption = "  Open to All "

End Sub

Sub Spin1_SpinUp ()
 Dim ct As Integer
  ct = (Val(LBL_CLIENT.Caption) + 1)
    If (ct > Val(MaxClientNumber)) Then ct = Val(MaxClientNumber)
     Else
         LBL_CLIENT.Caption = Str$(ct)
         Text_usr.Caption = "   " & CNAME$(ct)
```

```
End Sub

Sub Textl_MouseMove (Button As Integer, Shift As Integer, x As
Single, Y As Single)
   Online.Text = "Displays time "
End Sub

Sub Text2_MouseMove (Button As Integer, Shift As Integer,
x As Single, Y As Single)
End Sub

Sub Timerl_Timer ()
  KCTTIME.Caption = "        " + Time$
End Sub

Sub TXMSG_Change ()
 LBL_CHRLEN = Len(Txmsg.Text)
End Sub

Sub TXMSG_GotFocus ()
LBL_CHRLEN.Caption = Len(Txmsg.Text)
End Sub

VERSION 2.00
Begin Form PG_USER

Sub Btn_MaxExit_Click ()
 Unload PG_User
End Sub

Sub BtnMaxOK_Click ()
  SuperUser$ = Text_Max.Text
End Sub

VERSION 2.00
Begin Form PG_VNAME

Sub BtnVnameExit_Click ()
 BtnVnameList.Enabled = True
 Unload PG_VNAME
End Sub

Sub BtnVnameList_Click ()
list_vname.Clear
 FileNumber = FreeFile
 Open "C:\RAMESH\PAGER.VB\usrnam.DAT" For Input As FileNumber
 While Not EOF(FileNumber)
 Line Input #FileNumber, A$
```

```
 list_vname.AddItem A$
 Wend
 Close #FileNumber
 BtnVnameList.Enabled = False

End Sub

VERSION 2.00
Begin Form PG_VSMSG

Sub Command1_Click ()
list_vsmsg.Clear
 FileNumber = FreeFile
 Open "C:\RAMESH\PAGER.VB\stdmsg.DAT" For Input As FileNumber
 While Not EOF(FileNumber)
 Line Input #FileNumber, A$
list_vsmsg.AddItem A$
 Wend
 Close #FileNumber
 Command1.Enabled = False

End Sub

Sub Command2_Click ()
 Unload pg_vsmsg
End Sub


FILENAME: STDMSG.DAT


K.C.T Welcomes you ...
Have a Nice Day ...
Thank You Very Much
Happy New Year
Wish you A Grand Success
Happy Birthday to you
Merry Christmas
Come Home soon


USER NAME LIST

FILENAME: USRNAM.DAT
```

1. Dr.K.A.Palaniswamy
2. V.Chandrasekaran
3. V.Duraiswamy
4. S.Kumar
5. N.Kalaiarasi
6. T.Rani
7. SriKumar
8. S.Hariharan
9. SubbuLakshmi

5.4 OVER VIEW OF PL/M- 51 LANGUAGE

PL/M 51 is a high level language for programming Intel MCS-51 Family of embedded controllers. The language meets the software requirement of computers in a wide variety of systems and application work.

The PL/M 51 compiler is a software tool that translates PL/51 source code into a relocatable object module. These modules can then be combined with other modules coded in PL/M or other assembly languages. The complier provides printed listings error messages and number of controls that aid in developing and debugging programs .

The PL/M 51 complier can be installed on the IBM or IBM equivalent host system and used to generate object code for MCS 51 family of embedded controllers . The MCS 51 embedded controller family instruction sets are fully compatible.

APPLICATION DEVELOPMENT PROCESS

The PL/M 51 compliers and run time library are part and integrated set of tools that make of total MCS 51 development solution for embedded controller system.

5.4.1 STEPS IN SOFTWARE DEVELOPMENT PROCESS

1. Define the problem completely .
2. Outline the proposed solution in terms of hardware and software. Once the step is done ,begin defining the hardware.
3. Design the software for the system .

This important step consist of several sub steps including breaking down task into module , choosing the programing language and selecting the algorithms to be used .

4. Code the program and prepare them for translation using a text editor.

5. Translate PL/M program code using PL/M 51 compiler.

6. Using the text editor , correct any compile time error , then recompile.

7. Create and maintain libraries of commonly used objects modules Lib 51.

8. Link resulting relocatable object modules , and locate the object code using RL 51. When linking a program for a library, RL 51 extracts only the library modules that the program needs .

9. If PROM programming is used prepare the code for programming into PROM with OH .

10. Test the resulting program using symbolic debugger or other tools

CHAPTER 6

CONCLUSION

This project is simple low cost paging system which can be efficiently used for communication in Offices, Hospitals, Industries etc.

In the transmitter end an user friendly application software is written in visual basic which creates a menu so that the message can be sent to concerned person without any confusion.

This paging system is supported by the micro controller **8751** which has in-built EPROM. This makes the receiver unit very compact.

This paging system can also be used for covering greater distances by using five or more element YAGI ANTENNA instead of 1/4 wave length antenna as used in this project. This arrangement is shown in the figure (6.1). Thus with the help of the Yagi antenna, the distance covered can be increased to 0.5 K.M.