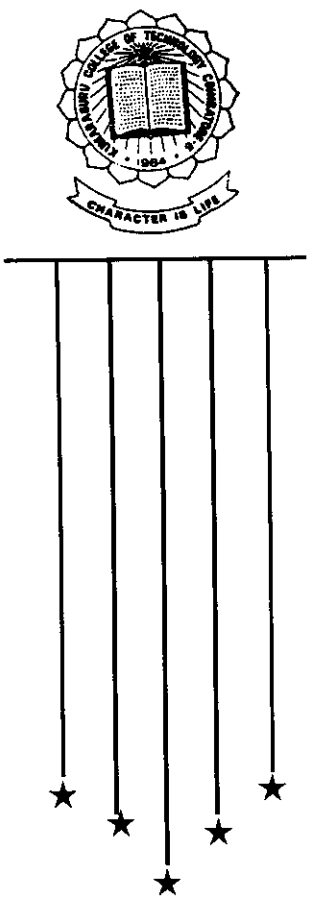
# WIRELESS DIGITAL DATA (IMAGE) TRANSMISSION

**PROJECT REPORT**



P-1369


*SUBMITTED BY*

**AGNEL JOSEPH**

**R. BALAJI**

**KANCHANA RAMASWAMY**

**RAMYA SRIDHAR**

*GUIDED BY*

**Prof. M. RAMASAMY,** M.E., M.I.S.T.E., M.I.E.E.E., M.I.E., (Engg)


IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE OF

**BACHELOR OF ENGINEERING IN**

**ELECTRONICS & COMMUNICATION ENGINEERING**

OF BHARATHIAR UNIVERSITY

**2000 - 2001**


***Department of Electronics & Communication Engineering***

***Kumaraguru College of Technology***

***Coimbatore - 641 006.***

# Kumaraguru College Of Technology

Coimbatore - 641 006

## Department of Electronics and Communication Engineering

## Certificate

This is to certify that this project entitled

**Wireless Digital Data (Image) Transmission**

Has been submitted by

Ms/Mr. [redacted]

In partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in the Electronics and Communication Engineering Branch of Bharathiar University, Coimbatore - 641 046 during the Academic year 2000-01

(Guide) (Head of the Department)

Certified that the candidate was examined by us in the Project Work.

Viva-Voce Examination held on 12.03.2001

University Register Number

.....

(Internal Examiner) (External Examiner)

Soliton
True engineering

## CERTIFICATE

## TO WHOMSOEVER IT MAY CONCERN

This is to certify that ***Agnel Joseph, Balaji R.,Kanchana Ramaswamy, Ramya Sridhar*** from the ECE department of Kumaraguru College of Engineering during the academic year 2000-2001 have worked on the project titled ***"Digital Wireless Data (Image) Transmission"*** as a part of their curriculum for the degree of Bachelor of Engineering in our organization & have demonstrated a fully working prototype of the same. They have shown a lot of interest & taken initiative in seeing this project through successfully & we wish them the very best in all their future endeavors.

Coimbatore –18,
03.03.2001

Anand P. Chinnaswamy
Director & CTO

**Soliton Automation India Pvt. Ltd.,** Classic Towers, 1547 Trichy Road, Coimbatore 641 018, INDIA
Tel : 91 422 302374, 302371 • Fax : 91 422 302375 • E-mail : mail@solitonautomation.com
Web : http://solitonautomation.com

# Acknowledgement

# *Synopsis*

# CONTENTS

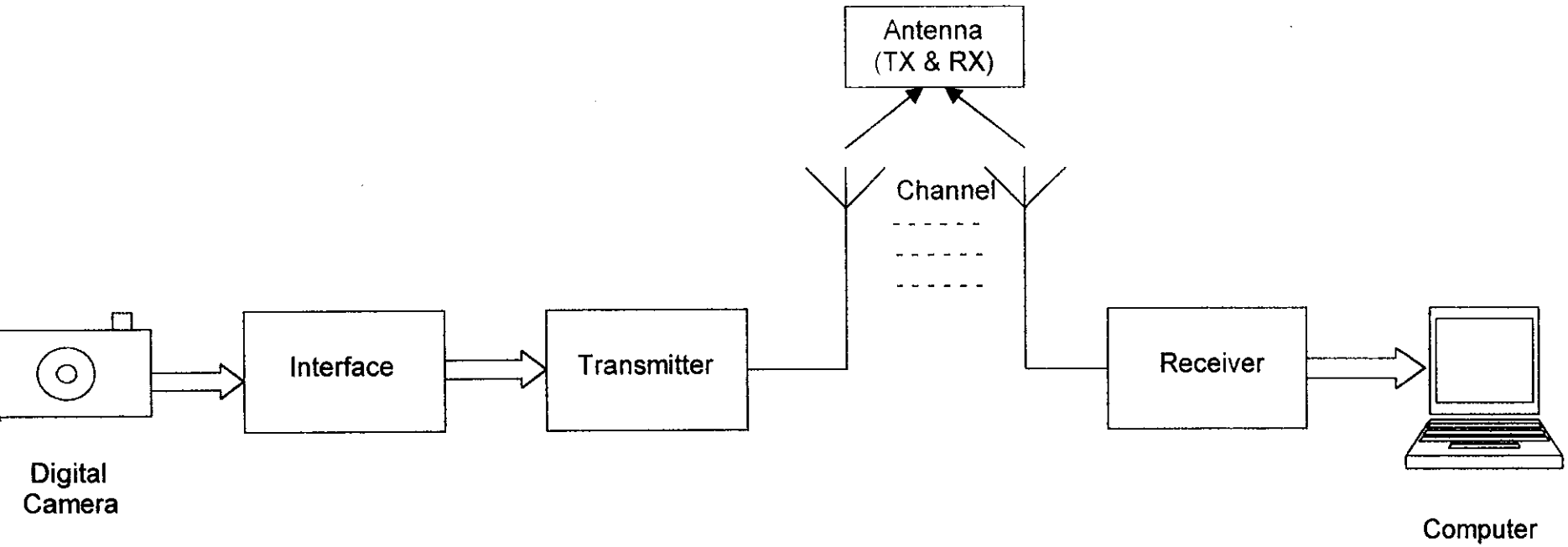
# INTRODUCTION

With vast improvements made in the field of communication everyday, the project "***Wireless digital image transmission***" deals with certain related issues like overcoming the capacity limitations of a floppy disc/memory card, that is commercially available, and facilitating a completely wireless link between a mobile and a stationary unit.

The block diagram of the project is shown in figure 1.

To communicate between the devices, various techniques were considered such as Bluetooth, wireless LAN etc., A detailed study of the above methods were undertaken and the results of the study are elaborated in the later sections. Taking into account various factors, it is found that frequency shift keying (FSK) is more suitable. The communication link between the micro controller and a PC has been established using FSK modulation of the digital data at the transmitter and subsequent demodulation at the receiver. The data is being transmitted and received using the serial ports of the micro controller and the PC respectively. The wireless link was established and tested at 110 bauds.

Intensive analysis and study has helped in the materialization of a working prototype model. The thesis presentation follows.




**GENERAL BLOCK DIAGRAM**

Fig 1

# BLUETOOTH

Bluetooth wireless technology is a defacto standard as well as a specification for small form factor, low cost, short-range radio links between mobile PC's, mobile phones and other portable devices. The bluetooth ***SIG*** (SPECIAL INTEREST GROUP) is a group, consisting leaders in the telecommunication and computing industries, that is driving development of technology and bringing into market Bluetooth protocol architecture.

Unlike IR (Infra Red), Bluetooth does not require line of sight positioning of connected units .The technology uses modification of existing Wireless LAN techniques but is most notable for its small size and low cost .Typical applications of bluetooth include e-mail delivery to laptops via phones potentially without user being aware that activity is taking place .

## BLUETOOTH PROTOCOL STACK

Different applications run on top of different application protocols. Consequently most applications use different set of protocols to achieve the desired functionality of the bluetooth usage models. These different set of protocols form the vertical slices of the bluetooth protocol stack with a common data link and physical layer. Additional vertical slices are for

services supportive of the main application like ***SOP*** (Service Discovery Protocol).

## THE APPLICATION

By connecting two bluetooth enabled devices data can be transmitted easily. In the project, the digital camera and the user interface (here a PC) have to be bluetooth enabled. To develop bluetooth enabled devices ***BDK*** (Bluetooth Development Kit) is required both at the transmitter and the receiver. The cost of this kit proved to be a limitation…..

**COST OF EACH BDK……$5800**

So, another affordable but equally efficient means of wireless communication technique is to be chosen. This leads to wireless LAN.

# WIRELESS LAN

A wireless local area network is a flexible data communications system implemented as an extension to or as an alternative for wired LAN. Wireless LAN combines data connectivity with user mobility. Wireless LAN users have a range of technologies to choose from when designing a wireless LAN solution. These are as follows.....

## 1) NARROWBAND TECHNOLOGY

A narrowband technology transmission system transmits and receives user information on a specific radio frequency. Narrowband radio keeps the radio signal frequency as narrow as possible just to pass information. Undesirable cross talk between communication channels is avoided by carefully coordinating different users on different channel frequencies.

## 2) SPREAD SPECTRUM TECHNOLOGY

Most Wireless LAN systems use this technology. This system produces a trade-off such that the signal is louder and easy to detect. Spread spectrum is designed to tradeoff bandwidth efficiency for reliability, integrity and security. There are two types of spread spectrum namely :

a) ***FREQUENCY HOPPING SPREAD SPECTRUM***

FHSS uses a narrowband carrier that changes frequency in a pattern known to both transmitter and receiver.

b) ***DIRECT SEQUENCE SPREAD SPECTRUM***

DSSS generates a redundant bit pattern for each bit to be transmitted. This bit pattern is called chirping code.

**3) INFRARED TECHNOLOGY**

This technology is used in commercial Wireless LAN .IR systems use very high frequencies just below visible light in the electromagnetic spectrum to carry data. IR waves cannot penetrate opaque objects .It is either ***directed*** (line –of- sight) or ***diffused*** technology. Directed systems provide very limited range (3ft) and typically are used for personal area networks.

**HOW WIRELESS LAN WORKS?**

Wireless LAN's use electromagnetic airwaves (radio or infrared) to communicate from one point to another without relying on any physical connection .The data being transmitted is superimposed on the radio carrier so

that it can be accurately extracted at the receiver end. This is referred to as modulation. To extract data, a radio receiver tunes in to one radio frequency while rejecting all other frequencies.

## RANGE AND COVERAGE

The distance over which RF and IR waves communicate is a function of product design and propagation path, especially indoor environments. Interactions with building walls, metal even human body can affect how energy propagates and thus what range and coverage a particular system achieves. Solid objects block infrared signals, which imposes additional limitations. Most Wireless LAN systems use RF, which can propagate through most obstacles.

## INTERNATIONAL WIRELESS LAN STANDARDS

Mobile frequency range ---------- 2400 –248.3 MHz

Multiple Access Methods --------- CSMA

Duplex Method ---------- TDD

Number of channels ---------- FHSS=79;DSSS=11

Channel spacing FHSS =1 MHz; DSSS=11 MHz



Data signals transmitted through pin 3 of the 9 pin connector of RS 232

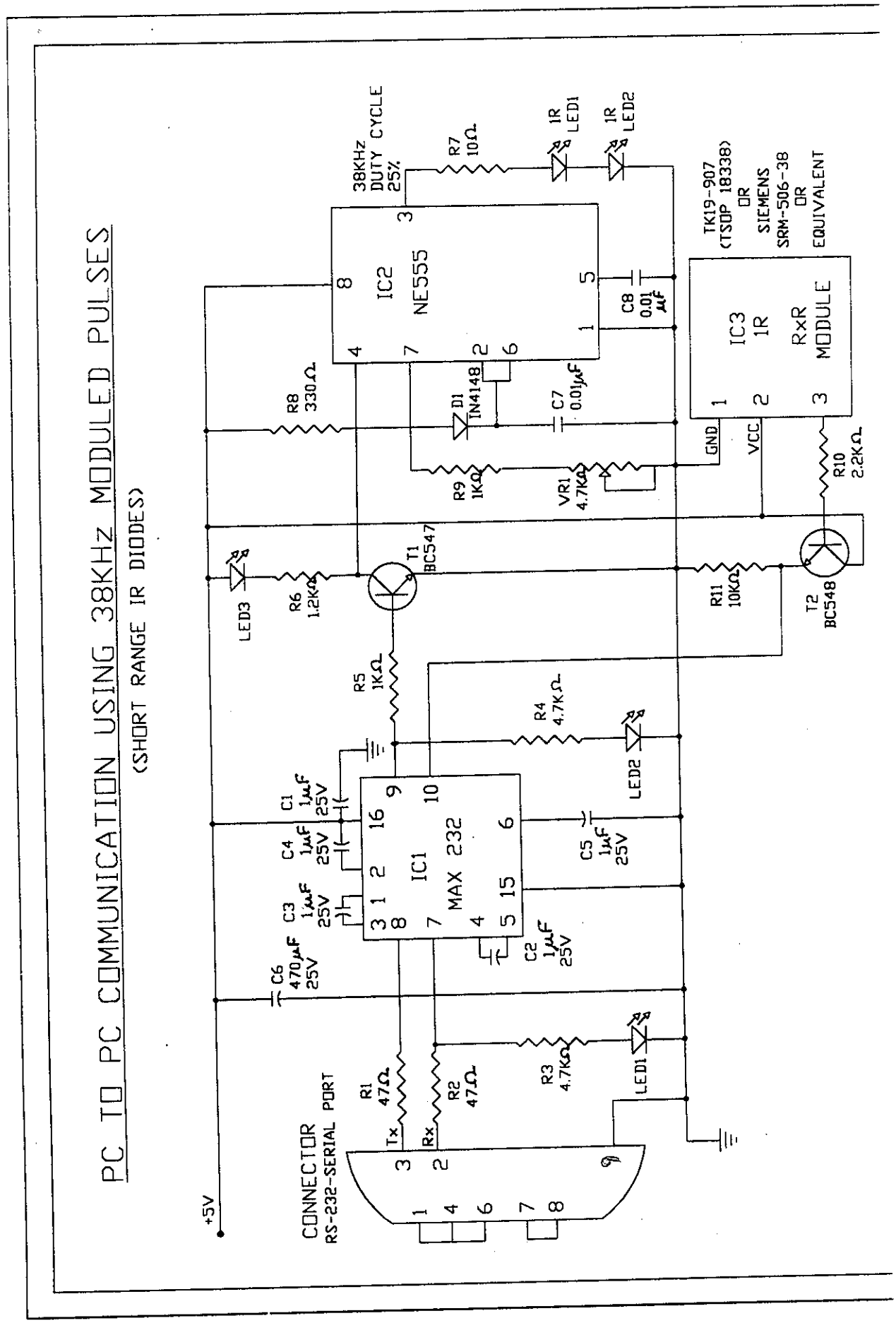

PC TO PC COMMUNICATION USING 38KHz MODULED PULSES
(SHORT RANGE IR DIODES)
38KHz DUTY CYCLE 25%
IC2 NE555
R7 10Ω
IR LED1
IR LED2
C8 0.01 µF
R8 330Ω
D1 IN4148
C7 0.01µF
R9 1KΩ
VR1 4.7KΩ
TK19-907 (TSOP 18338) OR SIEMENS SRM-506-38 OR EQUIVALENT
IC3 IR RxR MODULE
GND
VCC
R10 2.2KΩ
LED3
R6 1.2KΩ
T1 BC547
R5 1KΩ
R11 10KΩ
T2 BC548
R4 4.7KΩ
LED2
C1 1µF 25V
C4 1µF 25V
C3 1µF 25V
IC1 MAX 232
C5 1µF 25V
C2 1µF 25V
C6 470µF 25V
R1 47Ω
R2 47Ω
R3 4.7KΩ
LED1
Tx
Rx
CONNECTOR RS-232-SERIAL PORT
+5V

Fig. 2.

electric pulses sent by the COM port are now converted to corresponding modulated pulses of IR light.

## RECEIVER

The IR signals are detected by a photodiode(D1).The detected TTL levels are coupled to pin 10 of MAX 232 IC and are converted to + or - 9 V internally and output at pin 7.A visible LED at pin 7 indicates that the signals are being received.Pin 7 is also connected to pin 2 of the RS232 IC so that the data may be read.The optical signals received by the photo diodes are in fact converted to electrical pulses and thus reception is established.

## TESTING

Data transmission and reception were verified by flickering of indicative LED's when pulses were fed at pin 3 of RS 232 connector.The transmission and reception were verified on both the prototypes, one connected to each PC.

# WIRELESS LAN USING RADIO WAVES

Due to the disadvantages of IR transmission mentioned previously the next method was tested– using radio waves to transmit data .This was done using Frequency Shift Keying. The detailed report of this study is as below.

**TRANSMITTER**

The transmitter section consists of an FSK section and a high frequency carrier generator.

**Frequency Shift Keying:**

Frequency shift keying is a method of passing binary data in a signal. It is one of the most common methods used for transmitting digital data.

FSK takes binary data and creates a signal with changing frequency based on that data. A binary value of *zero* will cause the signal to use certain frequency and duty cycle for a period of time and a binary signal of *one* will cause the signal to use different frequency and duty cycle for a period of time.

In the project we have used the XR2206 IC to generate FSK.

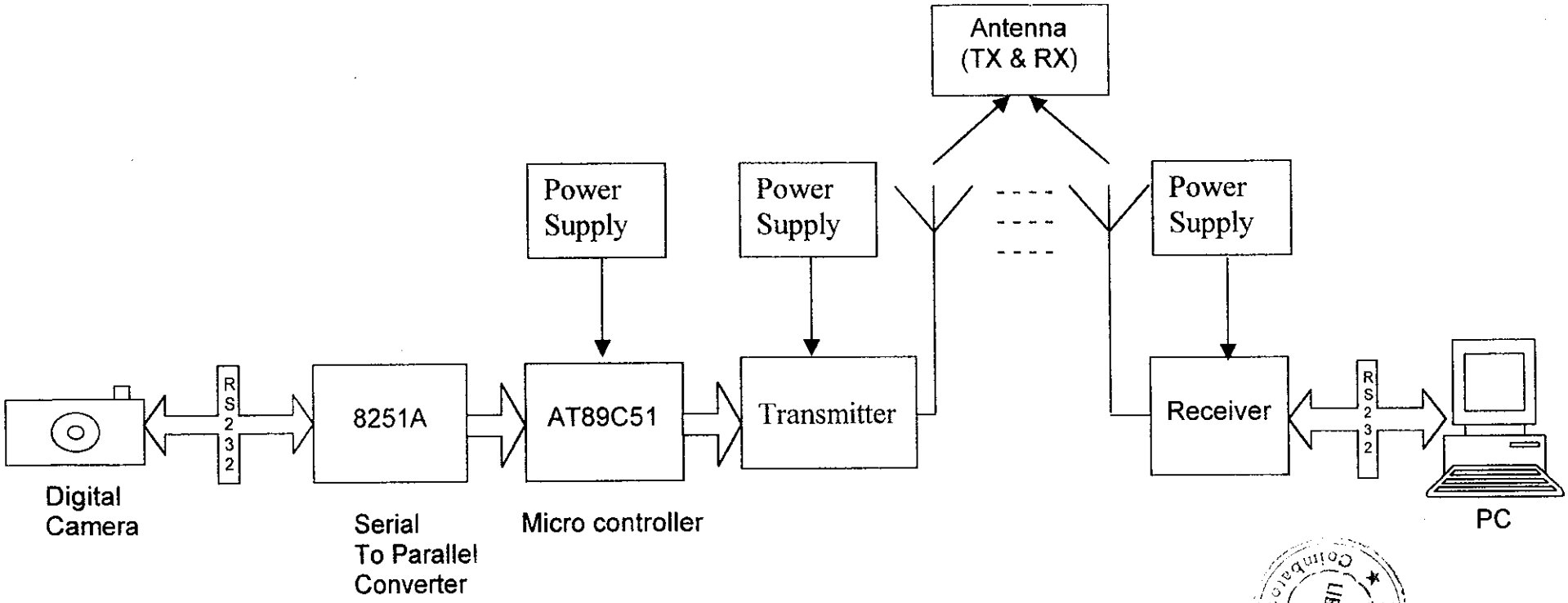



Fig 3. **DETAILED BLOCK DIAGRAM**

**XR2206 Function generator:**

The XR2206 is a monolithic function generator capable of producing high quality sine, square, triangle and ramp, and pulse waveforms of high-stability and accuracy .The output waveforms can be amplitude or frequency modulated by an external voltage. Frequency of operation can be selected externally over a range of 0.01Hz to more than 1MHz.The circuit can be used for FSK, AM and FM generation.

**FSK generation:**

The XR-2206 is comprises of four functional blocks; a voltage controlled oscillator, an analog multiplier and sine shaper, a unity gain current amplifier and a set of current switches. The VCO actually produces an output frequency proportional to the input current, which is produced from the timing terminals to ground, by a resistor. An FSK input pin, to produce an output frequency, controls the current to the VCO. With two timing pins, two discreet output frequencies can be independently produced for FSK generation applications. The circuit for using XR2206 as an FSK generator is shown in figure 4.

The XR-2206 can be operated with two separate timing resistors, $R_1$ and $R_2$, connected to the timing pins 7 and 8,respectively. Depending on the

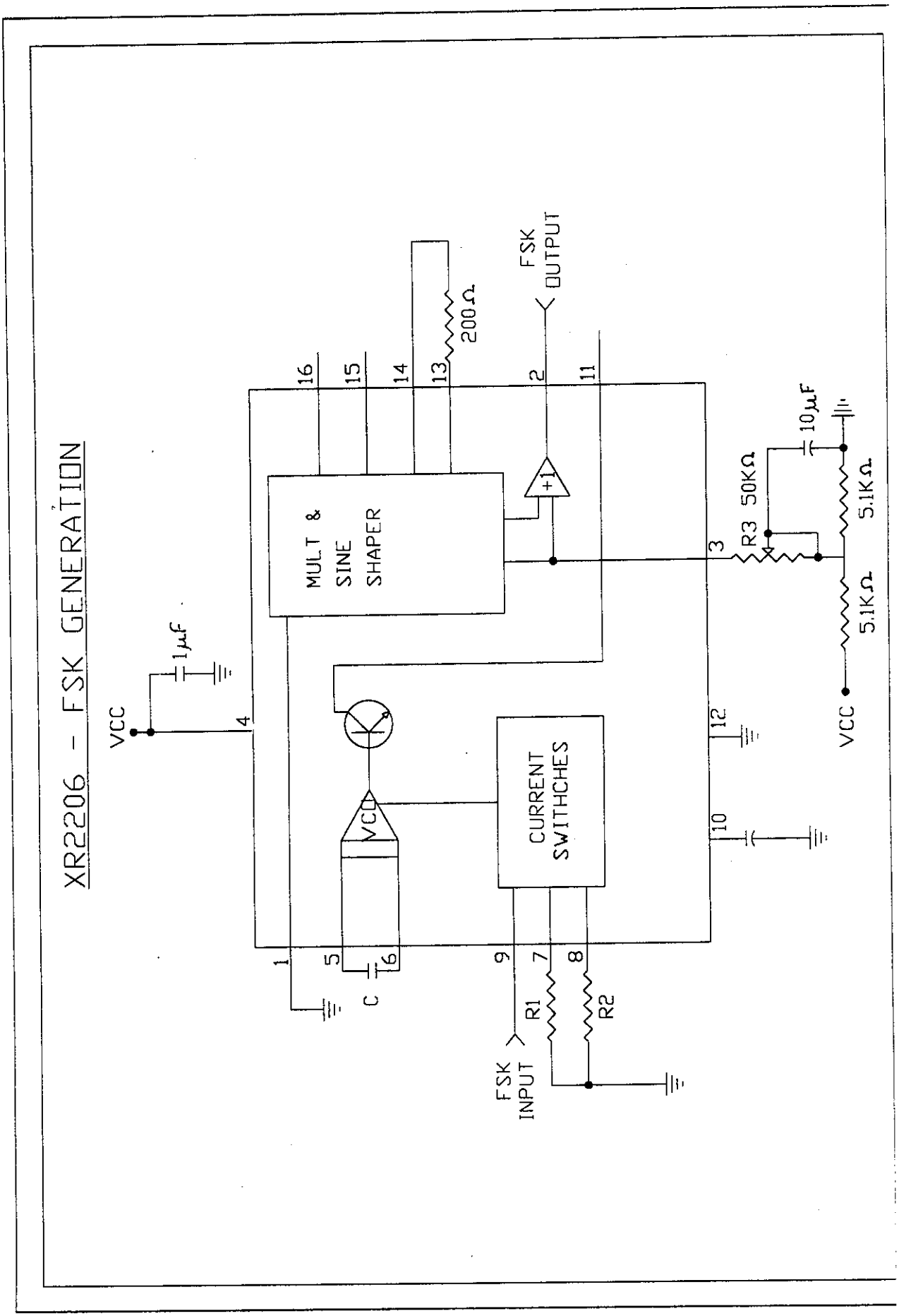

XR2206 - FSK GENERATION
MULT & SINE SHAPER
CURRENT SWITHCHES
VCO
+1
200Ω
FSK OUTPUT
FSK INPUT
VCC
1µF
10µF
50KΩ
5.1KΩ
5.1KΩ
R1
R2
R3
C
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Fig. 4.

polarity of the logic signal at Pin 9, either one or the other of these timing resistors is activated.

If Pin 9 is open-circuited or connected to a bias voltage –2V, only $R_1$ is activated. Similarly, if the voltage level at Pin 9 is –1V, only $R_2$ is activated. Thus, the output frequency can be keyed between two levels. $f_1$ and $f_2$,

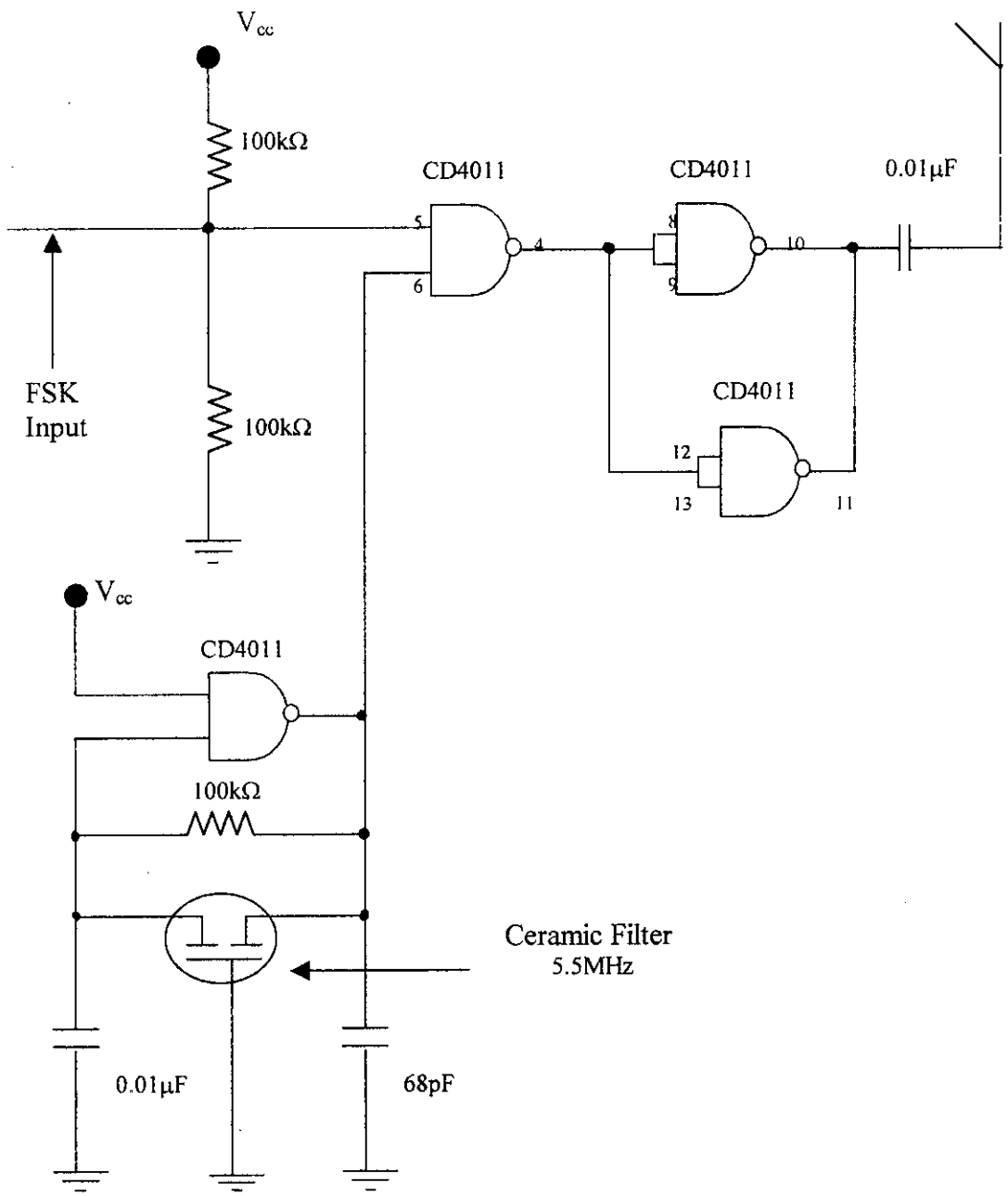
as:

$f_1 = 1/R_1 C$ and $f_2 = 1/R_2 C$

For split-supply operation, the keying voltage at Pin 9 is referenced to -V.

# CARRIER GENERATION

One input of the NAND gate is always maintained at a voltage of +12V. The other input is initially zero, as the capacitor is not charged. The output of the NAND gate is high (One). This causes the capacitor to charge through the resistor. The input at pin 2 of the NAND gate becomes a one and the output becomes zero. This process repeats to produce an oscillating output whose maximum frequency is limited to 5.5 MHz by a ceramic filter. This oscillating output is fed as an input to a second NAND gate. The other input of this NAND gate is the FSK output from XR2206. The output of the second NAND gate is therefore a modulated waveform. The Carrier generation schematic is shown in figure 5.




## CARRIER GENERATION

Fig 5.

# MICRO CONTROLLER

The difference between a microprocessor and micro controller is that a microprocessor can only process the data while a micro controller can control external devices. That is if you want switch "ON" or "OFF" a device, you need peripheral ICs to do this work but with a micro controller you can directly control the device.

Like a microprocessor, micro controller is available with different features. It is available with inbuilt memory, I/O lines, timer and ADC. The micro controller, used is 89C51, manufactured by Atmel, MC, USA. This is an advanced version of 8031. This micro controller has an inbuilt 4K bytes of flash ROM, 256 bytes of RAM, 32 I/O lines (4 bit ports) and 6 vectored interrupts.

## FLASH ROM

A 4-kilo byte ROM is available in the micro controller. It can be erased and reprogrammed. If the available memory is not enough for your program, you can interface the external ROM with this IC, it has 16 address lines, so a maximum of ($2^{16}$) i.e., 64 bytes of ROM can be interfaced with this Micro controller. Both internal and external ROM cannot be used simultaneously.

For external accessing of ROM, a pin is provided in the micro controller itself, i.e. pin no.31 EA should be high to use internal ROM, low to use external ROM

**RAM**

An internal 256 bytes of RAM is available for the user. These 256 bytes of RAM can be used along with the external RAM. Externally you can connect 64-kilo bytes of RAM with the micro controller. In internal RAM first 128 bytes of RAM is available for the user and the remaining 128 bytes is used as special function registers (SFR). These SFR's are used as control registers for timer, serial port etc.

**INPUT / OUTPUT PORTS**

There are four I/O ports available in AT89C51. They are port 0, port 1, port 2, port 3. All these ports are eight bit ports. All these ports can be controlled as eight-bit port or it can be controlled individually. One of the main features of this micro controller is that it can control the port pins individually. For example to control a LED we need to use one I/O line in the micro processor with 8255, we have to use an eight bit port. In 89C51 port 1 is available for users Port 3 is combined with interrupts. This can be used as interrupts (or) I/O ports, ports 2 & port 0 are combined with address bus & data bus.

All these port lines are available with internal pull-ups except port 0. If we want to use port 0 as I/O port we have to use pull up resistors.

This micro controller is working at a speed of maximum 24MHz. This micro controller is available with an inbuilt oscillator; we just have to connect the crystal to its terminal.

## ASSEMBLY LANGUAGE PROGRAM

The assembly language program is given below for serial communication between the micro controller and the PC.

```
;
;-----------------------------------
; Serial input option
;-----------------------------------
;
sin:        lcall   dsrc            ; display S
;
                mov     dpl,sdpl        ; get start addr
                mov     dph,sdph
                lcall   init_ser        ; initialize serial port
                mov     r0,#5           ; for 5 3f
;
se_bak:
                mov     a,scon          ; chk ri bit
                anl     a,#1
                cjne    a,#1,se_bak
;
                mov     a,sbuf          ; input data
                mov     scon,#58h       ; ready for next data
;
                cjne    a,#3fh,store    ; if data is not 3f, store
                djnz    r0,ser_skp      ; if r0 is not 0 .. Proceed
                ljmp    main
;
store:   mov     r0,#5
;
ser_skp:
                movx    @dptr,a         ; store
                inc     dptr
                ljmp    se_bak
```

```
;---------------------------------
; Serial output program
;---------------------------------
;
sout:       lcall   dsrc            ; display S
;
            lcall   dend            ; display E
;
            mov     dpl,sdpl        ; get src pointer
            mov     dph,sdph
;
            lcall   init_ser        ; initialize serial port
;
sobak:  movx    a,@dptr         ; get data from src
;
            mov     sbuf,a          ; output to sport
            lcall   delay
            lcall   delay
            mov     scon,#58h       ; ready for nxt data
;
            mov     a,dpl           ; compare current dpl with
edpl
            cjne    a,edpl,sopcd    ; proced
            mov     a,dph
            cjne    a,edph,sopcd
            ljmp    send_3f
;
sopcd:  inc     dptr
            ljmp    sobak
;
send_3f:
            mov     r0,#5
            mov     a,#3fh
tfbak:  mov     sbuf,a          ; output
            lcall   delay
            lcall   delay
            mov     scon,#58h       ; ready for nxt data
            djnz    r0,tfbak
            ljmp    main
;---------------------------------
; to display S at sixth ssd
;---------------------------------
dsrc:   lcall   blank_dis
            mov     dptr,#4001h
            mov     a,#90h
            movx    @dptr,a
            mov     dptr,#4000h
            mov     a,#1ch
            movx    @dptr,a
            lcall   get_addr        ; get addr
            mov     dptr,#ad_lo     ; save it in dpl & dph
            movx    a,@dptr
            mov     sdpl,a
            mov     dptr,#ad_hi
            movx    a,@dptr
            mov     sdph,a
            ret
;
;---------------------------------
; to display D at sixth ssd
;---------------------------------
```

```
ddst:    lcall   blank_dis
              mov     dptr,#4001h
              mov     a,#90h
              movx    @dptr,a
;
              mov     dptr,#4000h
              mov     a,#64h
              movx    @dptr,a

              lcall   get_addr        ; get addr
              mov     dptr,#ad_lo     ; save it in dpl & dph
              movx    a,@dptr
              mov     ddpl,a
              mov     dptr,#ad_hi
              movx    a,@dptr
              mov     ddph,a
              ret
;---------------------------------
; to display E at sixth ssd
;---------------------------------
dend:    lcall   blank_dis
              mov     dptr,#4001h
              mov     a,#90h
              movx    @dptr,a
              mov     dptr,#4000h
              mov     a,#16h
              movx    @dptr,a
;
              lcall   get_addr        ; get addr
;
              mov     dptr,#ad_lo     ; save it in dpl & dph
              movx    a,@dptr
              mov     edpl,a
              mov     dptr,#ad_hi
              movx    a,@dptr
              mov     edph,a
              ret
;-----------------------------------
; init serial port
;-----------------------------------
init_ser:
              mov     tmod,#20h       ; auto reload mode
              mov     th1,#0e6h       ; count
              mov     tcon,#40h       ; start
              mov     scon,#58h       ; receive enable and start
              lcall   delay
              ret


;-----------------------------------
; Routine to implement delay
;-----------------------------------
delay:   mov     r1,#0ffh
dl1:     mov     r2,#0ffh
dl2:     djnz    r2,dl2
             djnz    r1,dl1
             ret


;30h lsb,31h msb,254ahlsb,254bh msb
```

# TRANSMISSION AND RECEPTION

## TRANSMITTER WORKING

The transmitter circuit is as shown in figure 6.

The XR2206 is used to generate FSK depending on the signal level at pin 9. The input to the 9th pin of the XR2206 is the inverted data signal from the micro controller kit. When this signal is *zero* the transistor BC547 is cutoff, therefore the input to the 9th pin is +12 volts. When the signal from the micro controller is *one* the transistor BC547 is saturated so the net voltage at the 9th pin is 0 volts.

The voltage at the 9th pin controls the FSK generation. This determines the frequency ($f_1$, $f_2$) generated as mentioned previously .The frequency ranges from 1200Hz to 1500Hz.

The FSK output from pin 11 is fed to the modulation circuit. The other input to this circuit is the high frequency carrier signal. The output of this circuit is carrier superimposed on the FSK signal. This signal is fed to a buffer circuit .The buffer circuit is constructed using two CMOS NAND gates. This buffer is used to increase the current level of the signal to enable easy transmission using a wire. Hence the antenna used can be a wire.

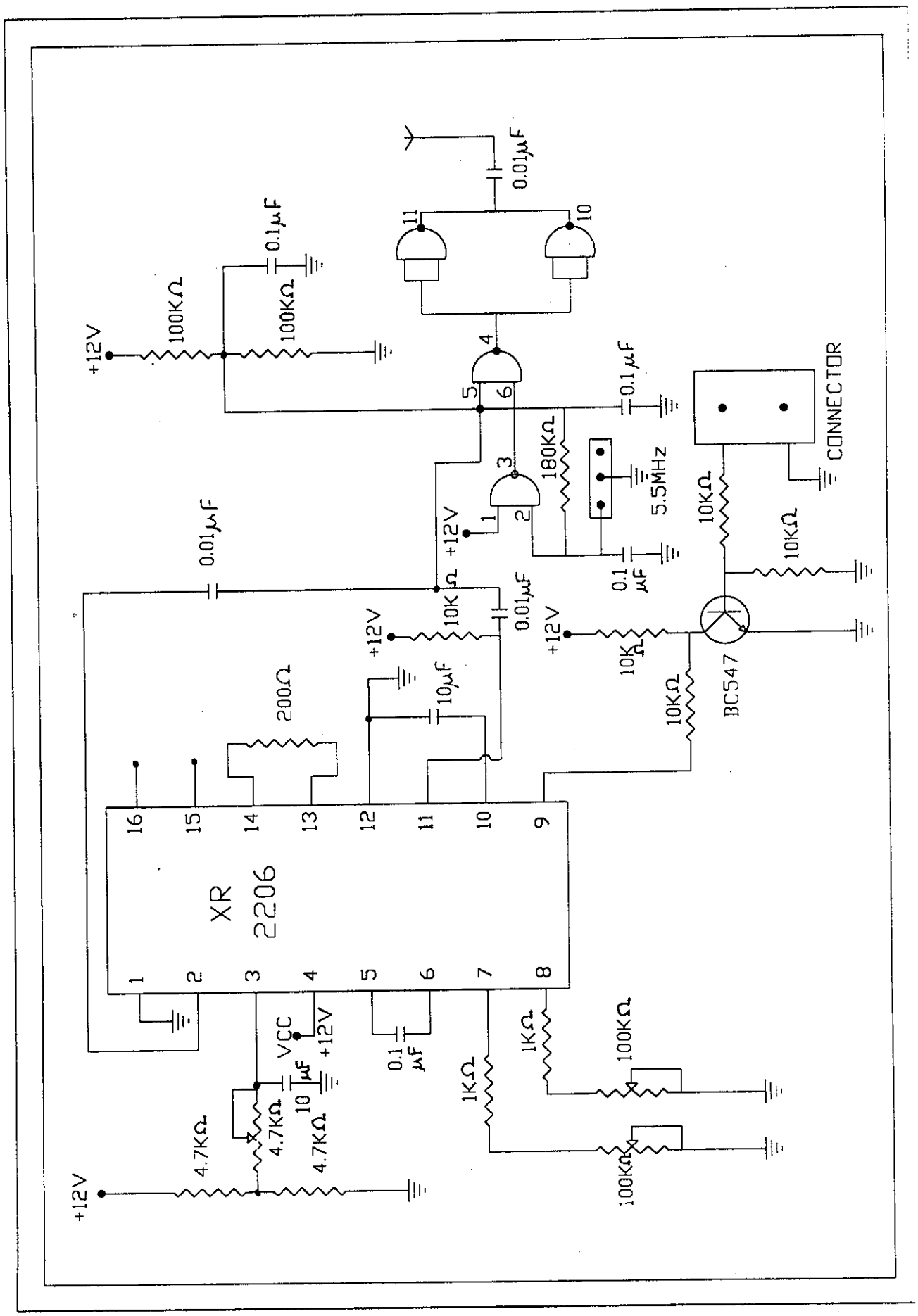



Fig 6. TRANSMITTER

## RECEIVER

The receiver circuit mainly comprises of RF amplifiers, filters and demodulation circuits. The wire antenna receives the data signal transmitted. This signal is fed to an RF amplifier constructed using a BFW10. This has high input impedance; hence it enables detection of weak signals. This followed by a ceramic filter. This filter removes any noise distortion and filters all signals above 5.5MHz.Then the signal is again fed to an amplifier and filter to improve the quality of the received signal.

The next stage is demodulation. For this an envelope detector circuit constructed using diode OA79 is used. This detector removes the carrier signal and outputs the pure FSK signal. This FSK is the required data signal. It is then amplified by a 741 opamp circuit .The amplified filtered signal is fed to the PLL 565.The details are given in figure 7.

## PLL 565:

565 is available as a 14 pin DIP package and as a 10 pin metal can package.

$$f_0=0.25/R_t.C_t \text{ Hz}$$

Where $R_t$ and $C_t$ are the resistor and capacitor connected to pin 8 & pin 9.The VCO free running frequency is adjusted to be at the centre of input

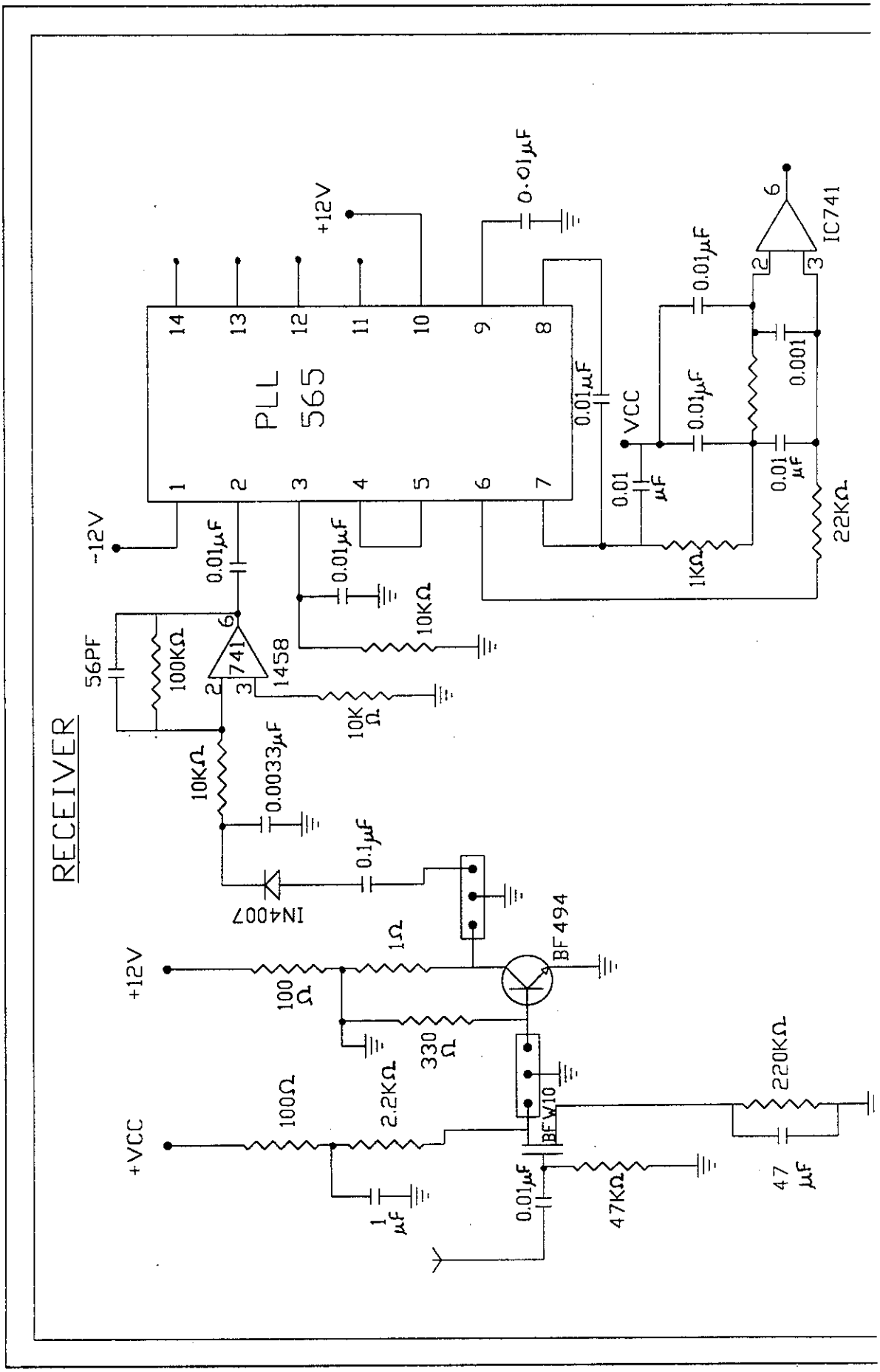

RECEIVER
PLL
565
+12V
-12V
+VCC
VCC
0.01µF
0.001
0.01 µF
22KΩ
1KΩ
IC741
741
1458
56PF
100KΩ
10KΩ
10K Ω
0.0033µF
0.1µF
IN4007
BF494
BFW10
100 Ω
1Ω
330 Ω
2.2KΩ
100Ω
220KΩ
47KΩ
47 µF
1 µF

Fig. 7.

frequency range. It can be seen that the free running frequency is internally broken between the VCO output and the phase comparator input.

A short circuit between pins 4 & 5 connect the VCO output to the phase comparator input so as to compare $f_0$ and input signal $f_s$. A capacitor C connected between pin7 and pin 10 to make a low pass filter with an internal resistance 3.6Kohm.

The lock in range is given by

$$\Delta f_L = \pm 7.8\ f0/V$$

The capture range is given by

$$[\Lambda f_L/2(pi)(3.6 \times 10^3)\ C]^{(1/2)}$$

Where C is in farads.

The capture range is located symmetrically with respect to VCO free running frequency $f_0$ as shown in the figure. The PLL cannot acquire a signal outside its capture range but once a signal is captured, it will hold on till the signal frequency goes beyond the lock in range .In order to increase the ability of the lock in range, a large capture range is required. This will however make the PLL more susceptible to noise and unwanted signals. Hence a suitable compromise is often reached between these two opposing requirements of the capture range. Many a time, the LPF bandwidth is first set as a large value for initial acquisition of the signal and once the signal is captured, the bandwidth of the LPF is reduced substantially. This will minimize the interference of undesirable signals and noise.

The output of the PLL is given to the comparator formed by the 741 IC. Depending on the frequency at pin 2 and 3, the voltage at pin 6 or 7 will dominate. Hence this will cause an output of +12V or –12V at the $6^{th}$ pin of the op-amp. The output of the op-amp is fed to the PC via the com1 port. The data from the com port is detected and displayed using software installed in the PC.

Two types of software were used . They are

1) XTALK
2) C program.

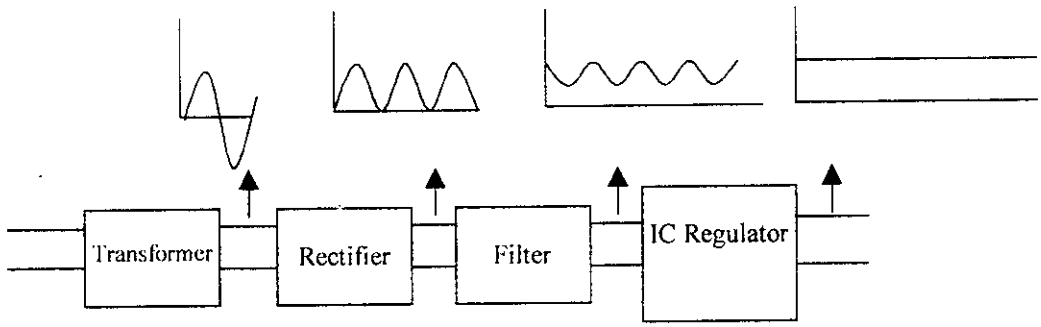



Fig 8.

## IC VOLTAGE REGULATORS

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection, all in a single IC. Although the internal construction of the IC is somewhat different from that described for discrete voltage regulator circuits, the external operation is much the same. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustably set voltage.

A power supply can be built using a transformer connected to the ac supply line to step the ac voltage to a desired amplitude, then rectifying that ac voltage, filtering with a capacitor and RC filter, if desired, and finally regulating the dc voltage using an IC regulator. The regulators can be selected for operation with load currents from hundreds of milli amperes to tens of amperes, corresponding to power ratings from milliwatts to tens of watts.

## THREE-TERMINAL VOLTAGE REGULATORS

Figure 9 shows the basic connection of a three-terminal voltage regulator IC to a load. The fixed voltage regulator has an unregulated dc input voltage, Vi, applied to one input terminal, a regulated output dc voltage, Vo, from a second terminal, with the third terminal connected to ground. For a selected regulator, IC device specifications list a voltage range over which the input voltage can vary to maintain a regulated output voltage over a range of load current. The specifications also list the amount of output voltage change resulting from a change in load current (load regulation) or in input voltage (line regulation).

**Fixed Positive Voltage Regulators:**

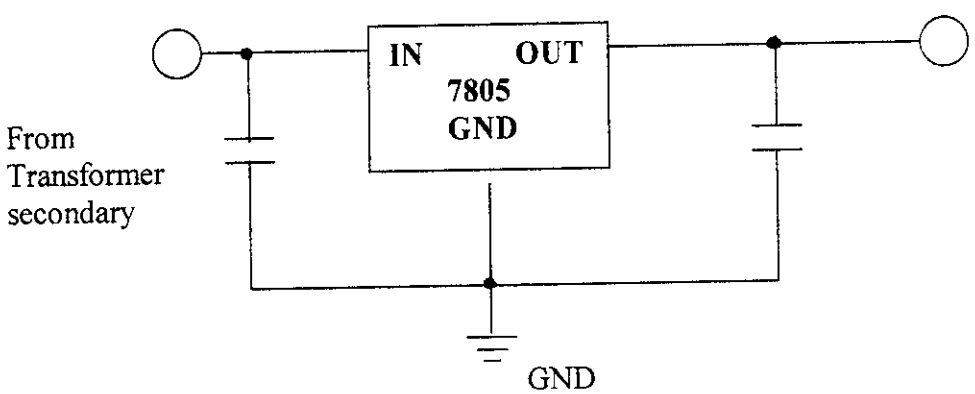



Fig 9.

The series 78 regulators provide fixed regulated voltages from 5 to 24 V. Figure 19.26 shows how one such IC, a 7812, is connected to provide voltage regulation with output from this unit of +12V dc. An unregulated input voltage Vi is filtered by capacitor C1 and connected to the IC's IN terminal. The IC's OUT terminal provides a regulated + 12V which is filtered by capacitor C2

(mostly for any high-frequency noise). The third IC terminal is connected to ground (GND). While the input voltage may vary over some permissible voltage range, and the output load may vary over some acceptable range, the output voltage remains constant within specified voltage variation limits. These limitations are spelled out in the manufacturer's specification sheets. A table of positive voltage regulated ICs is provided in table.

**TABLE: Positive Voltage Regulators in 7800 series**

| IC Part | OutputVoltage (V) | Minimum Vi (V) |
|---|---|---|
| 7805 | +5 | 7.3 |
| 7806 | +6 | 8.3 |
| 7808 | +8 | 10.5 |
| 7810 | +10 | 12.5 |
| 7812 | +12 | 14.6 |
| 7815 | +15 | 17.7 |
| 7818 | +18 | 21.0 |
| 7824 | +24 | 27.1 |

## POWER SUPPLY UNIT

## INTRODUCTION

Since all electronic circuits work only with low D.C. voltage we need a power supply unit to provide the appropriate voltage supply. This unit consists

of a transformer, rectifier, filter and a regulator. A.C. voltage typically 230V rms is connected to a transformer which steps that AC voltage down to the level to the desired AC voltage. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a DC voltage. This resulting DC voltage usually has some ripple or AC voltage variations. A regulator circuit can use this DC input to provide DC voltage that not only has much less ripple voltage but also remains the same DC value even the DC voltage varies some what, or the load connected to the output DC voltages changes.

## BLOCK DIAGRAM

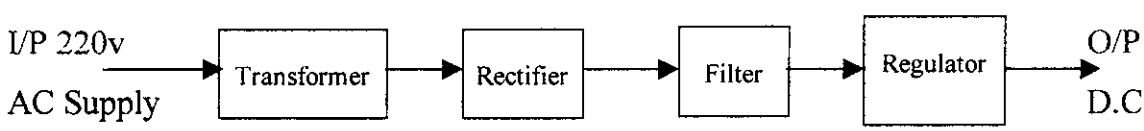



Fig 10.

## TRANSFORMER

A transformer is a static (or stationary) piece of which electric power in one circuit is transformed into electric power of the same frequency in another circuit. It can raise or lower the voltage in a circuit but with a corresponding decrease or increase in current. It works with the principle of mutual induction. In our project we are using step down transformer for providing the necessary supply for the electronic circuits. In our project we are using a 15-0-15 centre tapped transformer.

## RECTIFIER

The DC level obtained from a sinusoidal input can be improved 100% using a process called full-wave rectification. It uses 4 diodes in a bridge configuration. From the basic bridge configuration we see that two diodes (say D2 & D3) are conducting while the other two diodes (D1 & D4) are in "off" state during the period t =0 to T/2. Accordingly for he negative of the input the conducting diodes are D1 & D4. Thus the polarity across the load is the same.

## FILTER

The filter circuit used here is the capacitor filter circuit where a capacitor is connected at the rectifier output, and a DC is obtained across it. The filtered waveform is essentially a DC voltage with negligible ripples, which is ultimately fed to the load.

## REGULATOR

The output voltage from the capacitor is more filtered and finally regulated. The voltage regulator is a device, which maintains the output voltage constant irrespective of the change in supply variations, load variation and temperature changes. Here two fixed voltage regulators namely LM 7812, LM 7805 and LM7912 are used. The IC 7812 is a +12V regulator IC 7912 is a -12V regulator and IC 7805 is a +5V regulator.

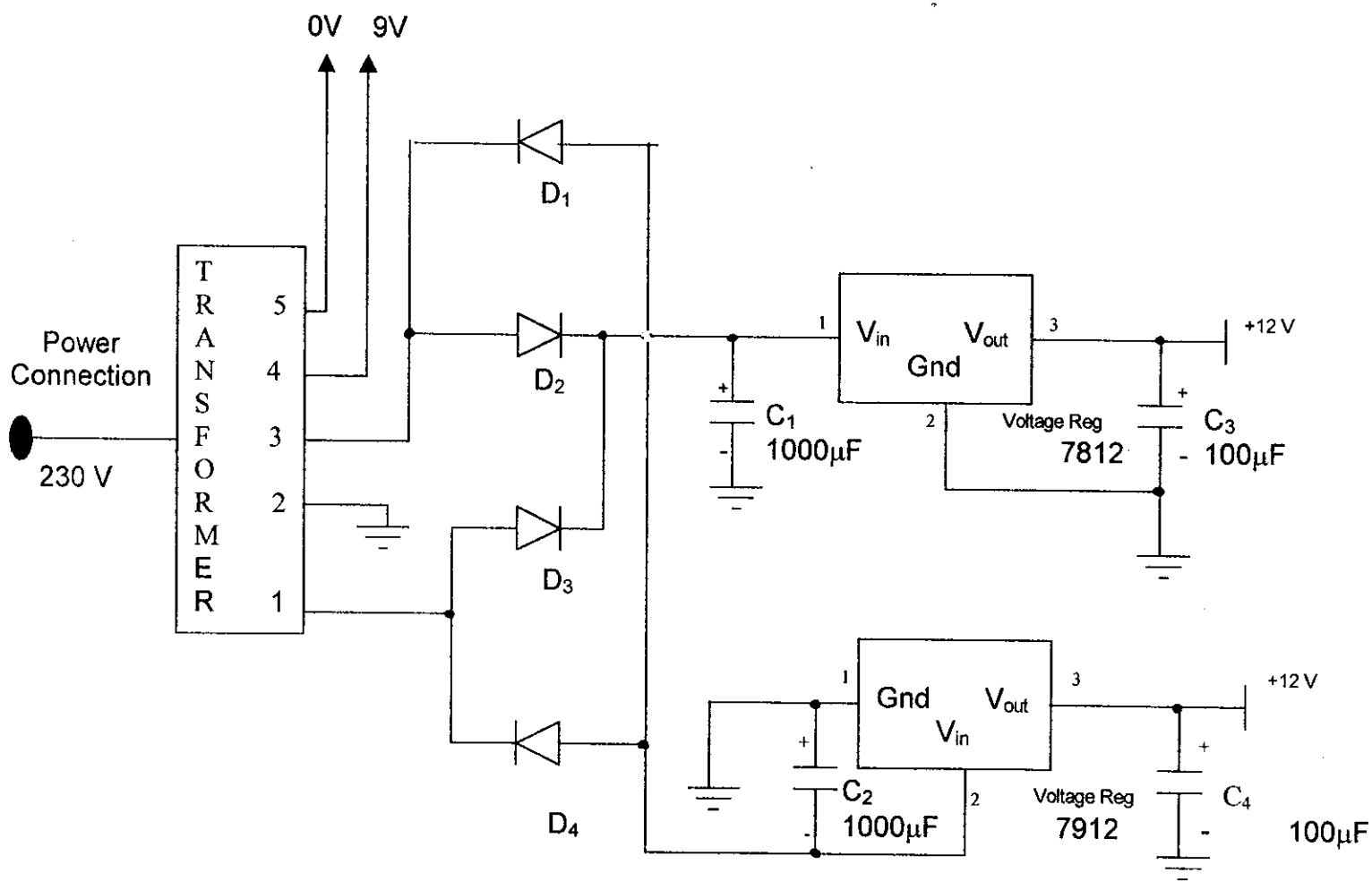



Fig 11. **POWER SUPPLY**

# INTERFACING CAMERA WITH THE MICRO CONTROLLER

The Digital Camera used here is one with a RS232 serial port. The camera output is to be fed to the distant workstation using the wireless link. For this a problem is encountered . The micro controller 89C51 has got only one serial port. This is used for the transmission of data to the receiver. The other port present is a parallel port. So the output from the camera coming out of the RS232 cable is to be converted to parallel data. This conversion is done using the 8251A serial to parallel converter.

The 8251A is a programmable chip designed for synchronous and asynchronous serial data communication, packaged in a 28-pin DIP. The 8251A is the enhanced version of its predecessor, the 8251. It includes five sections: Read/Write Control Logic, Transmitter, Receiver, Data Bus Buffer, and Modem Control.

The control logic interfaces the chip with the MPU, determines the functions of the chip according to the control word in its register, and monitors the data flow .The transmitter section converts a parallel word received from the MPU into serial bits and transmits them over the TxD line to a peripheral. The receiver section receives serial bits from a peripheral, converts them into a parallel word, and transfers the word to the MPU .The modem control is used to establish data communication through modems over telephone lines. The

8251A is a complex device, capable of performing various functions. The schematic of the interfacing circuit is shown in figure 12.

## TUNING

One of the most crucial areas in establishing wireless communication between the micro controller and PC is tuning. Tuning is done at the transmitter, in order to obtain two predetermined frequencies to represent '1' and '0'. In the receiver, at pin 3 of the PLL565 IC , the frequency is preset and this preset frequency is compared with the incoming frequency. Hence, there is no need for tuning at the receiver.

On the transmitter side , potentiometers are used in series with 1 K ohm resistors at pins numbers 7 and 8 in order to facilitate variation of the resistance. These resistance values along with the capacitor value of 0.1 micro farad connected between pins 5 and 6 serve to determine the frequencies used in FSK to represent the two binary signals , namely '1' and '0' .

*The relation between the FSK frequencies and the resistance values are given by*

$$F=1/(R*C)$$

*where R represents the effective resistance at pin 7 or 8 and*
*C represents the capacitance connected between pins 5 and 6*

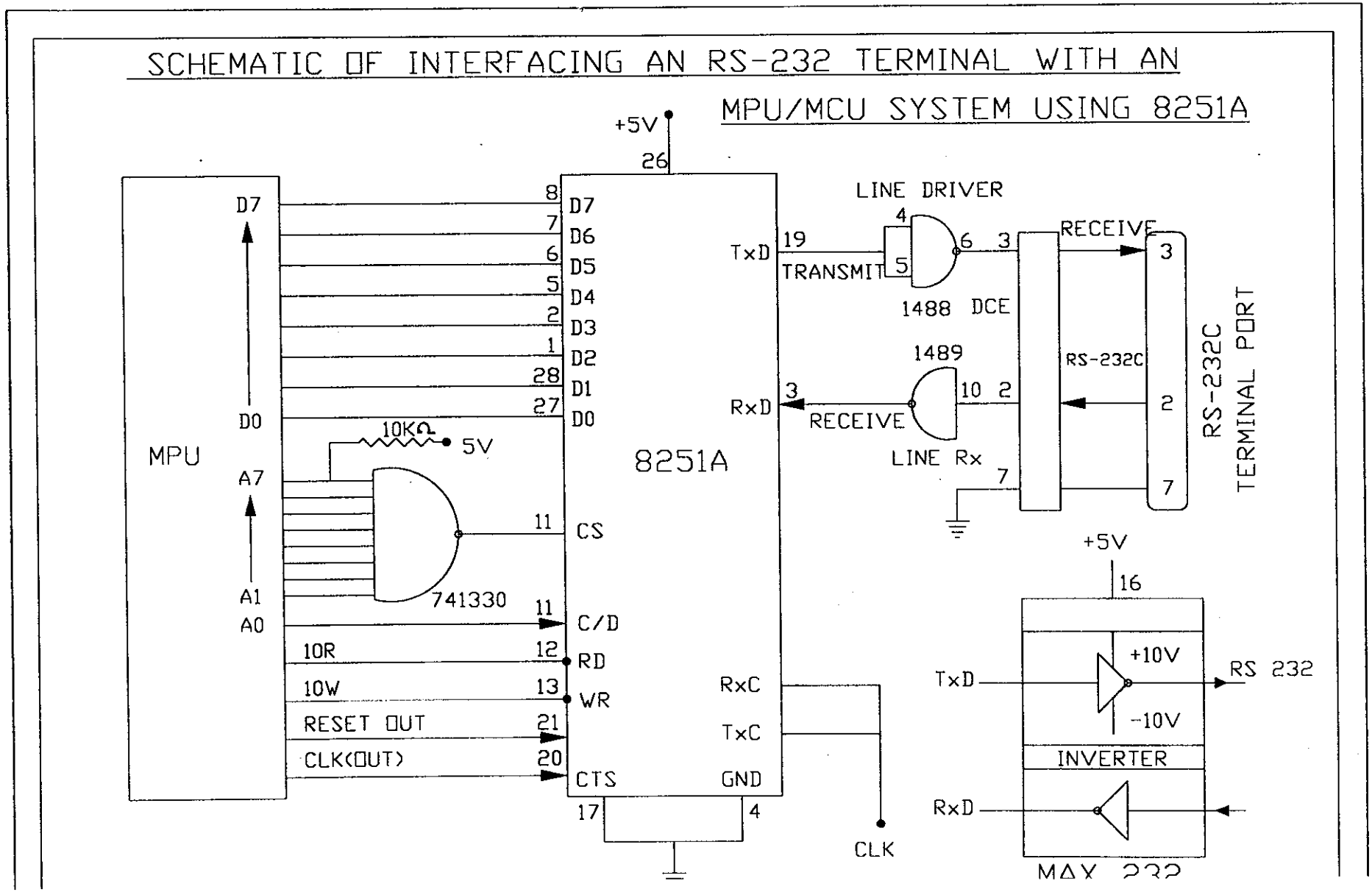

SCHEMATIC OF INTERFACING AN RS-232 TERMINAL WITH AN MPU/MCU SYSTEM USING 8251A
+5V
26
MPU
D7
D0
A7
A1
A0
8
7
6
5
2
1
28
27
D7
D6
D5
D4
D3
D2
D1
D0
8251A
10KΩ
5V
741330
11
CS
11
C/D
10R
12
RD
10W
13
WR
RESET OUT
21
CLK(OUT)
20
CTS
17
4
GND
RxC
TxC
CLK
LINE DRIVER
TxD
19
TRANSMIT
4
5
6
1488 DCE
3
RECEIVE
1489
RxD
3
RECEIVE
10
2
LINE Rx
7
RS-232C
3
2
7
RS-232C TERMINAL PORT
+5V
16
TxD
+10V
-10V
RS 232
INVERTER
RxD
MAX 232

Fig. 12.

Practically, tuning is achieved by viewing the received signal at pin 6, i.e. at the output of the IC741 used as a comparator and simultaneously tuning the transmitter in order to achieve a noise free reception .

The tuning of the transmitter is done by adjusting one potentiometer to give +12 V and another to give _12 V when the CD4011 IC's first pin is connected to the supply and ground respectively. Tuning is done until the noise signals that appeared in the form of pulses at the receiver, are eliminated.

Tuning is stopped when the received signal on the cathode ray oscilloscope indicates dc levels of +12 V and – 12V. Thereafter error free transmission is checked by transmission and reception of hexadecimal data from the micro controller.

# TESTING

## TESTING USING XTALK

For testing purposes the XVI version of XTALK developed for IBM machines is used. The status screen shows communication parameters, filter settings, key settings and control settings. Initially the parameters for communication and filtering are to be set. The two modes used in Xtalk are CALL mode and ANSWER mode.

For the project the following settings are to be made:-

- SPEED =0110
- DEBUG =HEX
- MODE =CALL
- INFILTER=OFF
- OUTFILTER=OFF

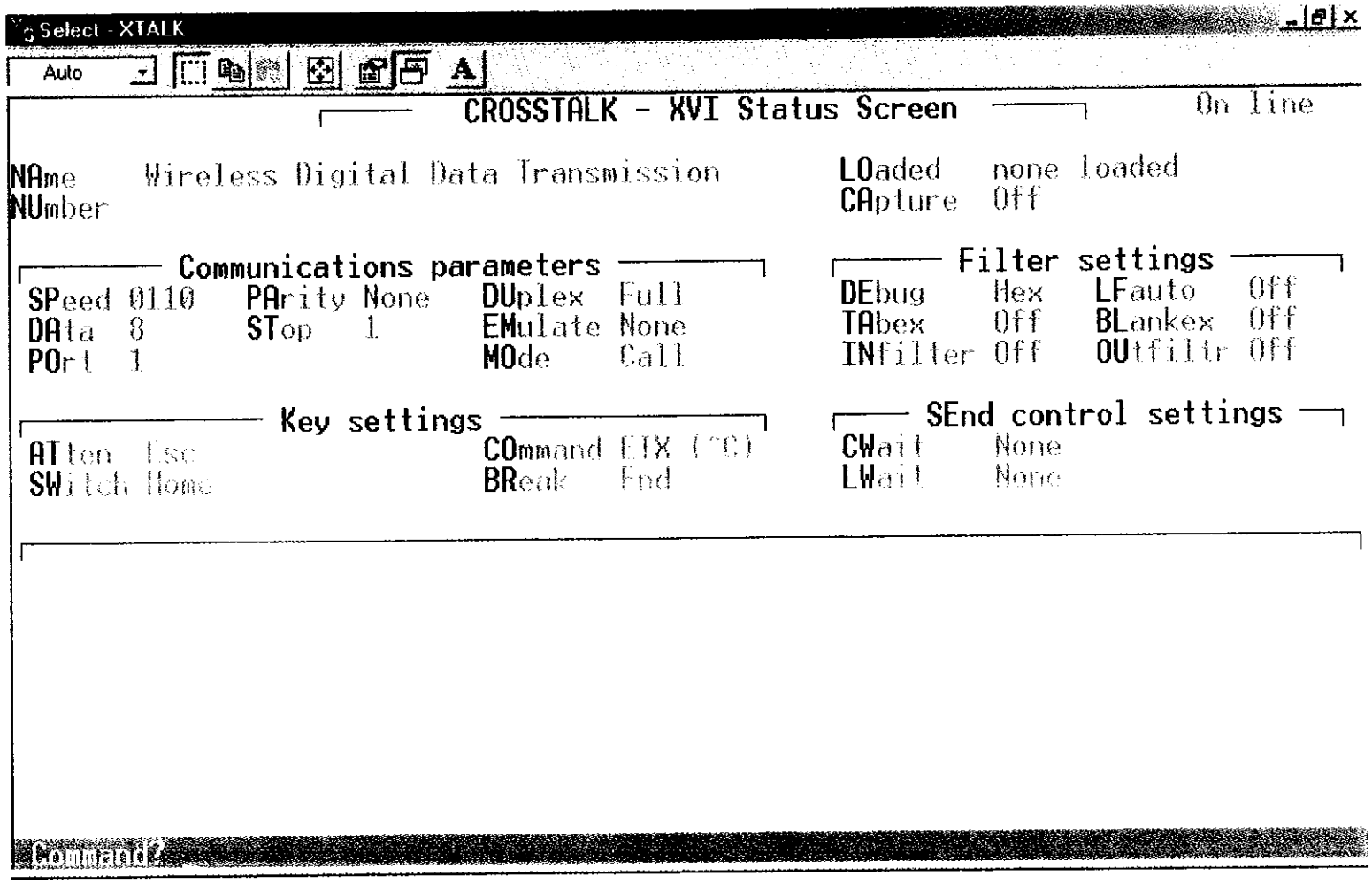



Fig 13. **XTALK HOME PAGE**

## TESTING USING C PROGRAM

The C program used for testing is given below. Before executing this program the COM port is to be initialized at the dos prompt using the command

**mode com1 110 n 8 1**

Here the mode at com1 port is set at a speed of 110 bauds, for 8 bit data and 1stop bit.

After this the C program is executed and the results are displayed in the output window.

The C program is as follows.

```
#include<process.h>
#include<stdio.h>
#include<conio.h>
#include<dos.h>
main()
{
  int i=0,x,y;
  system("mode com1,0110,n,8,1");
     for(;;)
{
      y=inportb(0x3f8);
      delay(800);
     printf("\n%x",y);
}

}
```

After initialization the command GO LOW is given. Then the local data link becomes active and the data, which is transmitted, will be displayed.

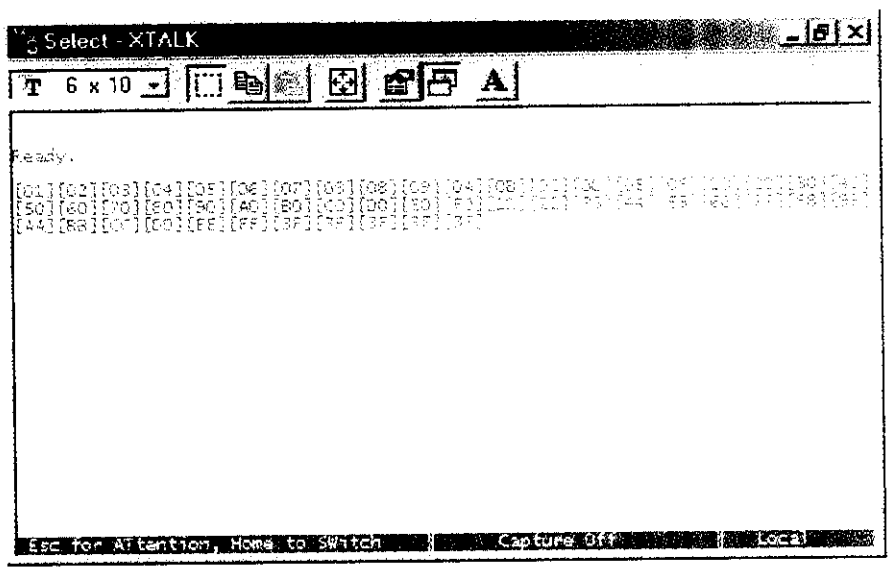



Fig 14. **RESULT SCREEN**

# FUTURE SCOPE

The final implementation of the project including the digital camera interface block would make it a truly novel one. The other developments that are visualized are,

1. Improving the frequency, speed and usage of high power transmitters and receivers can truly make this project a boon for all journalists.
2. Coding and realizing using VLSI techniques can be done to make the circuit truly handy.
3. Secrecy can be incorporated using advanced techniques like spread spectrum technology making the project viable for even defense applications.
4. Development of hardware to interface the camera with a cell phone (WAP enabled) can be done so that the image captured can be transferred to a WEB site.

# *CONCLUSION*

*This project was initiated to overcome the limitations of the memory card in a digital camera and enable unlimited photography, as the user is sure that the data will be stored in his workplace. This was possible using wireless communication technology.*

*Having successfully tested the data transmission over the wireless link between a micro controller and a PC , the concept stands vindicated. It is hoped that the project will be a beacon light leading to better developments in the relatively unexplored field of wireless image transfer.*

# BIBILOGRAPHY


1. Microprocessor architecture, Programming, and applications With the 8085. — Ramesh S. Gaonkar
2. Microprocessors and Interfacing — Douglas V. Hall
3. Microprocessor based design — Michael Slater
4. The 8051 Microcontroller — Kenneth J. Ayala
5. Communication Systems — Taub and Schilling
6. Communication Systems — Kennedy
7. Modern Digital and Analog Communication Systems — B.P.Lathi
8. Digital Communication — Haykin

# Data Sheets

## Features

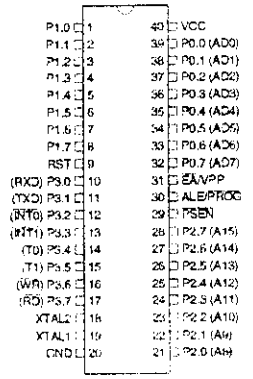
- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt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

## Description

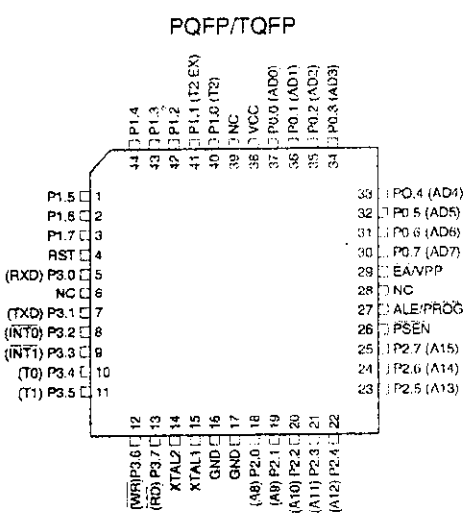
The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and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

## Pin Configurations

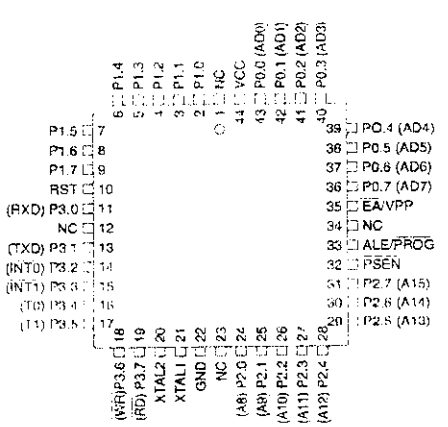
PDIP




PQFP/TQFP




PLCC

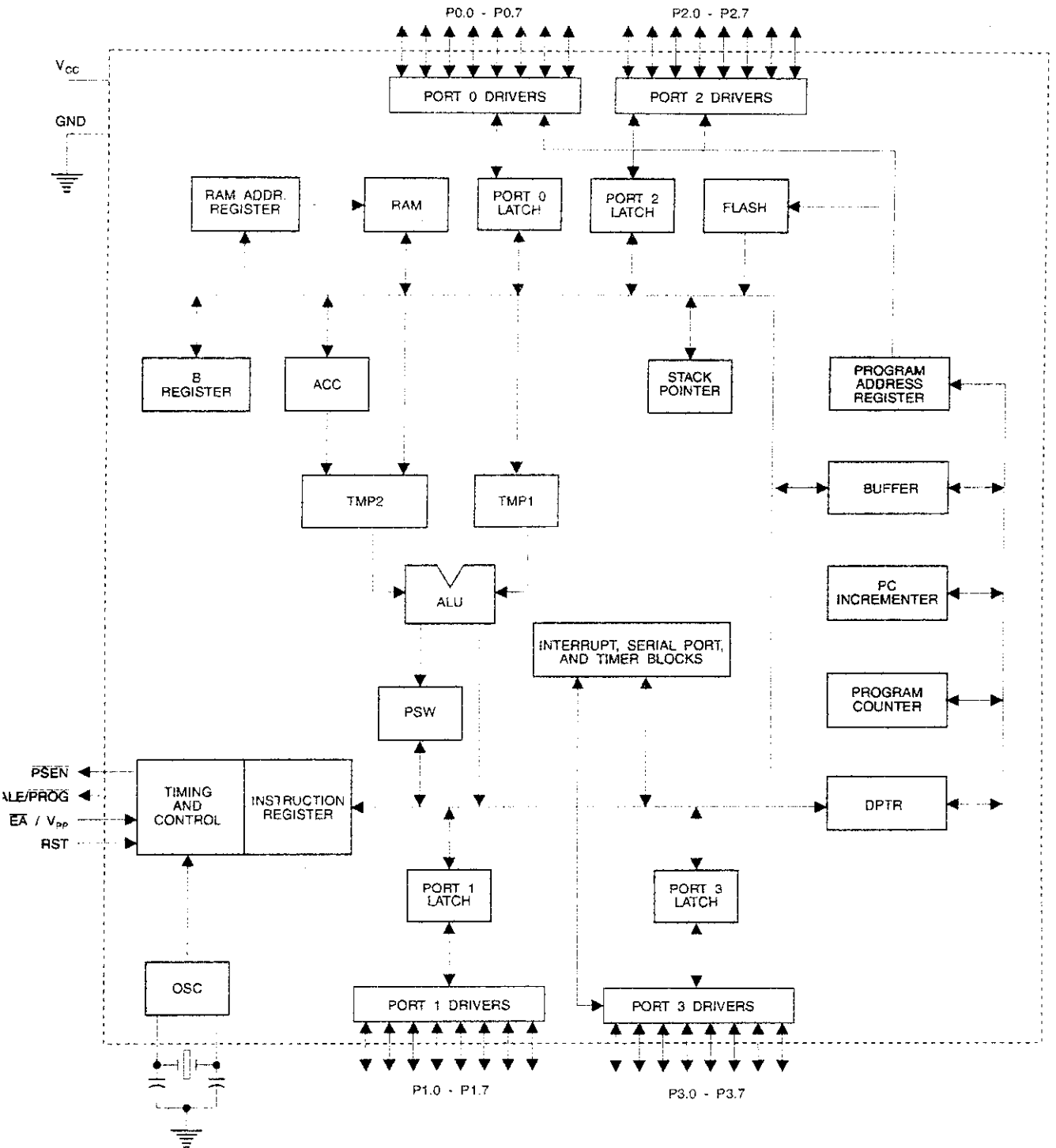



# 8-bit Microcontroller with 4K Bytes Flash

# AT89C51

Rev. 0265G–02/00

## Block Diagram



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

## Pin Description

### VCC

Supply voltage.

### GND

Ground.

### Port 0

Port 0 is an 8-bit open-drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

### Port 1

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

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

### Port 2

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

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

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

### Port 3

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

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

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

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

### RST

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

### ALE/$\overline{PROG}$

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

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

pulse is skipped during each access to external Data Memory.

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

**$\overline{\text{PSEN}}$**

Program Store Enable is the read strobe to external program memory.

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

**$\overline{\text{EA}}$/VPP**

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

$\overline{\text{EA}}$ should be strapped to $V_{CC}$ for internal program executions.

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

**XTAL1**

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

**XTAL2**

Output from the inverting oscillator amplifier.

## Oscillator Characteristics

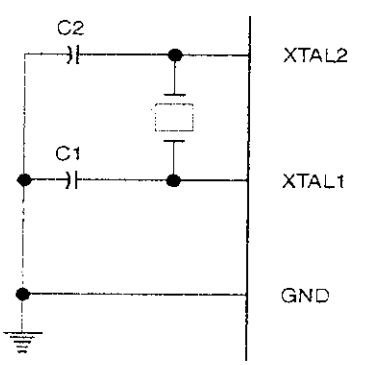
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected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

## Idle Mode

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

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

**Figure 1.** Oscillator Connections

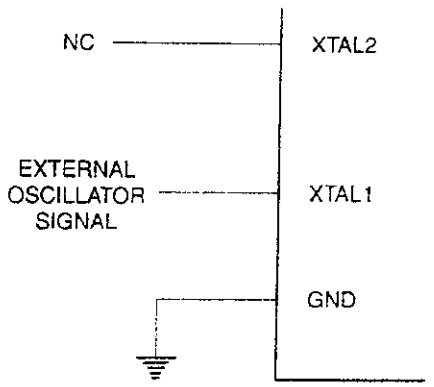



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

## Status of External Pins During Idle and Power-down Modes

| Mode | Program Memory | ALE | $\overline{\text{PSEN}}$ | PORT0 | PORT1 | PORT2 | PORT3 |
|---|---|---|---|---|---|---|---|
| Idle | Internal | 1 | 1 | Data | Data | Data | Data |
| Idle | External | 1 | 1 | Float | Data | Address | Data |
| Power-down | Internal | 0 | 0 | Data | Data | Data | Data |
| Power-down | External | 0 | 0 | Float | Data | Data | Data |

**Figure 2.** External Clock Drive Configuration




## Power-down Mode

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

## Program Memory Lock Bits

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

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

## Lock Bit Protection Modes

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

## Programming the Flash

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

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

| | $V_{PP}$ = 12V | $V_{PP}$ = 5V |
|---|---|---|
| Top-Side Mark | AT89C51<br>xxxx<br>yyww | AT89C51<br>xxxx-5<br>yyww |
| Signature | (030H) = 1EH<br>(031H) = 51H<br>(032H) =F FH | (030H) = 1EH<br>(031H) = 51H<br>(032H) = 05H |

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

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

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

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

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

**Program Verify:** If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

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

**Reading the Signature Bytes:** The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

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

## Programming Interface

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

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

| Mnemonic | | Description | Byte | Oscillator Period |
|---|---|---|---|---|
| RR | A | Rotate Accumulator Right | 1 | 12 |
| RRC | A | Rotate Accumulator Right through the Carry | 1 | 12 |
| SWAP | A | Swap nibbles within the Accumulator | 1 | 12 |
| **DATA TRANSFER** | | | | |
| MOV | A,$R_n$ | Move register to Accumulator | 1 | 12 |
| MOV | A,direct | Move direct byte to Accumulator | 2 | 12 |
| MOV | A,@$R_i$ | Move indirect RAM to Accumulator | 1 | 12 |
| MOV | A,#data | Move immediate data to Accumulator | 2 | 12 |
| MOV | $R_n$,A | Move Accumulator to register | 1 | 12 |
| MOV | $R_n$,direct | Move direct byte to register | 2 | 24 |
| MOV | $R_n$,#data | Move immediate data to register | 2 | 12 |
| MOV | direct,A | Move Accumulator to direct byte | 2 | 12 |
| MOV | direct,$R_n$ | Move register to direct byte | 2 | 24 |
| MOV | direct,direct | Move direct byte to direct | 3 | 24 |
| MOV | direct,@$R_i$ | Move indirect RAM to direct byte | 2 | 24 |
| MOV | direct,#data | Move immediate data to direct byte | 3 | 24 |
| MOV | @$R_i$,A | Move Accumulator to indirect RAM | 1 | 12 |
| MOV | @$R_i$,direct | Move direct byte to indirect RAM | 2 | 24 |
| MOV | @$R_i$,#data | Move immediate data to indirect RAM | 2 | 12 |
| MOV | DPTR,#data16 | Load Data Pointer with a 16-bit constant | 3 | 24 |
| MOVC | A,@A+DPTR | Move Code byte relative to DPTR to Acc | 1 | 24 |
| MOVC | A,@A+PC | Move Code byte relative to PC to Acc | 1 | 24 |
| MOVX | A,@$R_i$ | Move External RAM (8-bit addr) to Acc | 1 | 24 |
| **DATA TRANSFER (continued)** | | | | |
| MOVX | A,@DPTR | Move External RAM (16-bit addr) to Acc | 1 | 24 |
| MOVX | @$R_i$,A | Move Acc to External RAM (8-bit addr) | 1 | 24 |
| MOVX | @DPTR,A | Move Acc to External RAM (16-bit addr) | 1 | 24 |
| PUSH | direct | Push direct byte onto stack | 2 | 24 |
| POP | direct | Pop direct byte from stack | 2 | 24 |
| XCH | A,$R_n$ | Exchange register with Accumulator | 1 | 12 |
| XCH | A,direct | Exchange direct byte with Accumulator | 2 | 12 |
| XCH | A,@$R_i$ | Exchange indirect RAM with Accumulator | 1 | 12 |
| XCHD | A,@$R_i$ | Exchange low-order Digit indirect RAM with Acc | 1 | 12 |
| **BOOLEAN VARIABLE MANIPULATION** | | | | |
| CLR | C | Clear Carry | 1 | 12 |
| CLR | bit | Clear direct bit | 2 | 12 |
| SETB | C | Set Carry | 1 | 12 |
| SETB | bit | Set direct bit | 2 | 12 |
| CPL | C | Complement Carry | 1 | 12 |
| CPL | bit | Complement direct bit | 2 | 12 |
| ANL | C,bit | AND direct bit to CARRY | 2 | 24 |
| ANL | C,/bit | AND complement of direct bit to Carry | 2 | 24 |
| ORL | C,bit | OR direct bit to Carry | 2 | 24 |
| ORL | C,/bit | OR complement of direct bit to Carry | 2 | 24 |
| MOV | C,bit | Move direct bit to Carry | 2 | 12 |
| MOV | bit,C | Move Carry to direct bit | 2 | 24 |
| JC | rel | Jump if Carry is set | 2 | 24 |
| JNC | rel | Jump if Carry not set | 2 | 24 |
| JB | bit,rel | Jump if direct Bit is set | 3 | 24 |
| JNB | bit,rel | Jump if direct Bit is Not set | 3 | 24 |
| JBC | bit,rel | Jump if direct Bit is set & clear bit | 3 | 24 |
| **PROGRAM BRANCHING** | | | | |

**Table 1.** AT89 Instruction Set Summary[1]

| Mnemonic | | Description | Byte | Oscillator Period |
|---|---|---|---|---|
| **ARITHMETIC OPERATIONS** | | | | |
| ADD | A,$R_n$ | Add register to Accumulator | 1 | 12 |
| ADD | A,direct | Add direct byte to Accumulator | 2 | 12 |
| ADD | A,@$R_i$ | Add indirect RAM to Accumulator | 1 | 12 |
| ADD | A,#data | Add immediate data to Accumulator | 2 | 12 |
| ADDC | A,$R_n$ | Add register to Accumulator with Carry | 1 | 12 |
| ADDC | A,direct | Add direct byte to Accumulator with Carry | 2 | 12 |
| ADDC | A,@$R_i$ | Add indirect RAM to Accumulator with Carry | 1 | 12 |
| ADDC | A,#data | Add immediate data to Acc with Carry | 2 | 12 |
| SUBB | A,$R_n$ | Subtract Register from Acc with borrow | 1 | 12 |
| SUBB | A,direct | Subtract direct byte from Acc with borrow | 2 | 12 |
| SUBB | A,@$R_i$ | Subtract indirect RAM from ACC with borrow | 1 | 12 |
| SUBB | A,#data | Subtract immediate data from Acc with borrow | 2 | 12 |
| INC | A | Increment Accumulator | 1 | 12 |
| INC | $R_n$ | Increment register | 1 | 12 |
| INC | direct | Increment direct byte | 2 | 12 |
| INC | @$R_i$ | Increment direct RAM | 1 | 12 |
| DEC | A | Decrement Accumulator | 1 | 12 |
| DEC | $R_n$ | Decrement Register | 1 | 12 |
| DEC | direct | Decrement direct byte | 2 | 12 |
| DEC | @$R_i$ | Decrement indirect RAM | 1 | 12 |
| INC | DPTR | Increment Data Pointer | 1 | 24 |
| MUL | AB | Multiply A & B | 1 | 48 |
| DIV | AB | Divide A by B | 1 | 48 |
| DA | A | Decimal Adjust Accumulator | 1 | 12 |

Note: 1. All mnemonics copyrighted © Intel Corp., 1980.

| Mnemonic | | Description | Byte | Oscillator Period |
|---|---|---|---|---|
| **LOGICAL OPERATIONS** | | | | |
| ANL | A,$R_n$ | AND Register to Accumulator | 1 | 12 |
| ANL | A,direct | AND direct byte to Accumulator | 2 | 12 |
| ANL | A,@$R_i$ | AND indirect RAM to Accumulator | 1 | 12 |
| ANL | A,#data | AND immediate data to Accumulator | 2 | 12 |
| ANL | direct,A | AND Accumulator to direct byte | 2 | 12 |
| ANL | direct,#data | AND immediate data to direct byte | 3 | 24 |
| ORL | A,$R_n$ | OR register to Accumulator | 1 | 12 |
| ORL | A,direct | OR direct byte to Accumulator | 2 | 12 |
| ORL | A,@$R_i$ | OR indirect RAM to Accumulator | 1 | 12 |
| ORL | A,#data | OR immediate data to Accumulator | 2 | 12 |
| ORL | direct,A | OR Accumulator to direct byte | 2 | 12 |
| ORL | direct,#data | OR immediate data to direct byte | 3 | 24 |
| XRL | A,$R_n$ | Exclusive-OR register to Accumulator | 1 | 12 |
| XRL | A,direct | Exclusive-OR direct byte to Accumulator | 2 | 12 |
| XRL | A,@$R_i$ | Exclusive-OR indirect RAM to Accumulator | 1 | 12 |
| XRL | A,#data | Exclusive-OR immediate data to Accumulator | 2 | 12 |
| XRL | direct,A | Exclusive-OR Accumulator to direct byte | 2 | 12 |
| XRL | direct,#data | Exclusive-OR immediate data to direct byte | 3 | 24 |
| CLR | A | Clear Accumulator | 1 | 12 |
| CPL | A | Complement Accumulator | 1 | 12 |
| RL | A | Rotate Accumulator Left | 1 | 12 |
| RLC | A | Rotate Accumulator Left through the Carry | 1 | 12 |
| **LOGICAL OPERATIONS (continued)** | | | | |

| Mnemonic | | Description | Byte | Oscillator Period |
|---|---|---|---|---|
| ACALL | addr11 | Absolute Subroutine Call | 2 | 24 |
| LCALL | addr16 | Long Subroutine Call | 3 | 24 |
| RET | | Return from Subroutine | 1 | 24 |
| RETI | | Return from interrupt | 1 | 24 |
| AJMP | addr11 | Absolute Jump | 2 | 24 |
| LJMP | addr16 | Long Jump | 3 | 24 |
| SJMP | rel | Short Jump (relative addr) | 2 | 24 |
| JMP | @A+DPTR | Jump indirect relative to the DPTR | 1 | 24 |
| JZ | rel | Jump if Accumulator is Zero | 2 | 24 |
| JNZ | rel | Jump if Accumulator is Not Zero | 2 | 24 |
| CJNE | A,direct,rel | Compare direct byte to Acc and Jump if Not Equal | 3 | 24 |
| CJNE | A,#data,rel | Compare immediate to Acc and Jump if Not Equal | 3 | 24 |
| CJNE | $R_n$,#data,rel | Compare immediate to register and Jump if Not Equal | 3 | 24 |
| CJNE | @$R_i$,#data,rel | Compare immediate to indirect and Jump if Not Equal | 3 | 24 |
| DJNZ | $R_n$,rel | Decrement register and Jump if Not Zero | 2 | 24 |
| DJNZ | direct,rel | Decrement direct byte and Jump if Not Zero | 3 | 24 |
| NOP | | No Operation | 1 | 12 |

**Table 2.** Instruction Opcodes in Hexadecimal Order

| Hex Code | Number of Bytes | Mnemonic | Operands |
|---|---|---|---|
| 00 | 1 | NOP | |
| 01 | 2 | AJMP | code addr |
| 02 | 3 | LJMP | code addr |
| 03 | 1 | RR | A |
| 04 | 1 | INC | A |
| 05 | 2 | INC | data addr |
| 06 | 1 | INC | @R0 |
| 07 | 1 | INC | @R1 |
| 08 | 1 | INC | R0 |
| 09 | 1 | INC | R1 |
| 0A | 1 | INC | R2 |
| 0B | 1 | INC | R3 |
| 0C | 1 | INC | R4 |
| 0D | 1 | INC | R5 |
| 0E | 1 | INC | R6 |
| 0F | 1 | INC | R7 |
| 10 | 3 | JBC | bit addr,code addr |
| 11 | 2 | ACALL | code addr |
| 12 | 3 | LCALL | code addr |
| 13 | 1 | RRC | A |
| 14 | 1 | DEC | A |
| 15 | 2 | DEC | data addr |
| 16 | 1 | DEC | @R0 |
| 17 | 1 | DEC | @R1 |
| 18 | 1 | DEC | R0 |
| 19 | 1 | DEC | R1 |
| 1A | 1 | DEC | R2 |
| 1B | 1 | DEC | R3 |
| 1C | 1 | DEC | R4 |
| 1D | 1 | DEC | R5 |
| 1E | 1 | DEC | R6 |
| 1F | 1 | DEC | R7 |
| 20 | 3 | JB | bit addr,code addr |
| 21 | 2 | AJMP | code addr |
| 22 | 1 | RET | |
| 23 | 1 | RL | A |
| 24 | 2 | ADD | A,#data |
| 25 | 2 | ADD | A,data addr |

| Hex Code | Number of Bytes | Mnemonic | Operands |
|---|---|---|---|
| 26 | 1 | ADD | A,@R0 |
| 27 | 1 | ADD | A,@R1 |
| 28 | 1 | ADD | A,R0 |
| 29 | 1 | ADD | A,R1 |
| 2A | 1 | ADD | A,R2 |
| 2B | 1 | ADD | A,R3 |
| 2C | 1 | ADD | A,R4 |
| 2D | 1 | ADD | A,R5 |
| 2E | 1 | ADD | A,R6 |
| 2F | 1 | ADD | A,R7 |
| 30 | 3 | JNB | bit addr,code addr |
| 31 | 2 | ACALL | code addr |
| 32 | 1 | RETI | |
| 33 | 1 | RLC | A |
| 34 | 2 | ADDC | A,#data |
| 35 | 2 | ADDC | A,data addr |
| 36 | 1 | ADDC | A,@R0 |
| 37 | 1 | ADDC | A,@R1 |
| 38 | 1 | ADDC | A,R0 |
| 39 | 1 | ADDC | A,R1 |
| 3A | 1 | ADDC | A,R2 |
| 3B | 1 | ADDC | A,R3 |
| 3C | 1 | ADDC | A,R4 |
| 3D | 1 | ADDC | A,R5 |
| 3E | 1 | ADDC | A,R6 |
| 3F | 1 | ADDC | A,R7 |
| 40 | 2 | JC | code addr |
| 41 | 2 | AJMP | code addr |
| 42 | 2 | ORL | data addr,A |
| 43 | 3 | ORL | data addr,#data |
| 44 | 2 | ORL | A,#data |
| 45 | 2 | ORL | A,data addr |
| 46 | 1 | ORL | A,@R0 |
| 47 | 1 | ORL | A,@R1 |
| 48 | 1 | ORL | A,R0 |
| 49 | 1 | ORL | A,R1 |
| 4A | 1 | ORL | A,R2 |

| Hex Code | Number of Bytes | Mnemonic | Operands |
|---|---|---|---|
| 97 | 1 | SUBB | A,@R1 |
| 98 | 1 | SUBB | A,R0 |
| 99 | 1 | SUBB | A,R1 |
| 9A | 1 | SUBB | A,R2 |
| 9B | 1 | SUBB | A,R3 |
| 9C | 1 | SUBB | A,R4 |
| 9D | 1 | SUBB | A,R5 |
| 9E | 1 | SUBB | A,R6 |
| 9F | 1 | SUBB | A,R7 |
| A0 | 2 | ORL | C,/bit addr |
| A1 | 2 | AJMP | code addr |
| A2 | 2 | MOV | C,bit addr |
| A3 | 1 | INC | DPTR |
| A4 | 1 | MUL | AB |
| A5 | | reserved | |
| A6 | 2 | MOV | @R0,data addr |
| A7 | 2 | MOV | @R1,data addr |
| A8 | 2 | MOV | R0,data addr |
| A9 | 2 | MOV | R1,data addr |
| AA | 2 | MOV | R2,data addr |
| AB | 2 | MOV | R3,data addr |
| AC | 2 | MOV | R4,data addr |
| AD | 2 | MOV | R5,data addr |
| AE | 2 | MOV | R6,data addr |
| AF | 2 | MOV | R7,data addr |
| B0 | 2 | ANL | C,/bit addr |
| B1 | 2 | ACALL | code addr |
| B2 | 2 | CPL | bit addr |
| B3 | 1 | CPL | C |
| B4 | 3 | CJNE | A,#data,code addr |
| B5 | 3 | CJNE | A,data addr,code addr |
| B6 | 3 | CJNE | @R0,#data,code addr |
| B7 | 3 | CJNE | @R1,#data,code addr |
| B8 | 3 | CJNE | R0,#data,code addr |
| B9 | 3 | CJNE | R1,#data,code addr |
| BA | 3 | CJNE | R2,#data,code addr |
| BB | 3 | CJNE | R3,#data,code addr |
| BC | 3 | CJNE | R4,#data,code addr |

| Hex Code | Number of Bytes | Mnemonic | Operands |
|---|---|---|---|
| BD | 3 | CJNE | R5,#data,code addr |
| BE | 3 | CJNE | R6,#data,code addr |
| BF | 3 | CJNE | R7,#data,code addr |
| C0 | 2 | PUSH | data addr |
| C1 | 2 | AJMP | code addr |
| C2 | 2 | CLR | bit addr |
| C3 | 1 | CLR | C |
| C4 | 1 | SWAP | A |
| C5 | 2 | XCH | A,data addr |
| C6 | 1 | XCH | A,@R0 |
| C7 | 1 | XCH | A,@R1 |
| C8 | 1 | XCH | A,R0 |
| C9 | 1 | XCH | A,R1 |
| CA | 1 | XCH | A,R2 |
| CB | 1 | XCH | A,R3 |
| CC | 1 | XCH | A,R4 |
| CD | 1 | XCH | A,R5 |
| CE | 1 | XCH | A,R6 |
| CF | 1 | XCH | A,R7 |
| D0 | 2 | POP | data addr |
| D1 | 2 | ACALL | code addr |
| D2 | 2 | SETB | bit addr |
| D3 | 1 | SETB | C |
| D4 | 1 | DA | A |
| D5 | 3 | DJNZ | data addr,code addr |
| D6 | 1 | XCHD | A,@R0 |
| D7 | 1 | XCHD | A,@R1 |
| D8 | 2 | DJNZ | R0,code addr |
| D9 | 2 | DJNZ | R1,code addr |
| DA | 2 | DJNZ | R2,code addr |
| DB | 2 | DJNZ | R3,code addr |
| DC | 2 | DJNZ | R4,code addr |
| DD | 2 | DJNZ | R5,code addr |
| DE | 2 | DJNZ | R6,code addr |
| DF | 2 | DJNZ | R7,code addr |
| E0 | 1 | MOVX | A,@DPTR |
| E1 | 2 | AJMP | code addr |
| E2 | 1 | MOVX | A,@R0 |

March 1988

National Semiconductor

# CD4001BM/CD4001BC Quad 2-Input NOR Buffered B Series Gate
# CD4011BM/CD4011BC Quad 2-Input NAND Buffered B Series Gate

## General Description

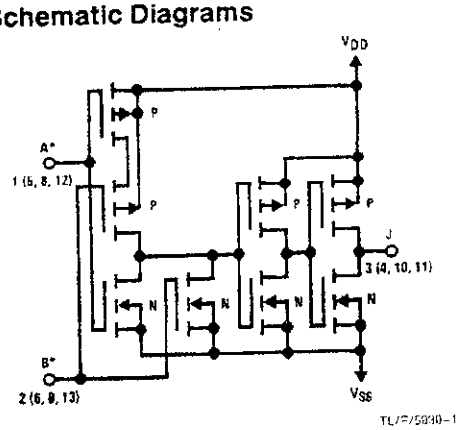
These quad gates are monolithic complementary MOS (CMOS) integrated circuits constructed with N- and P-channel enhancement mode transistors. They have equal source and sink current capabilities and conform to standard B series output drive. The devices also have buffered outputs which improve transfer characteristics by providing very high gain.

All inputs are protected against static discharge with diodes to $V_{DD}$ and $V_{SS}$.

## Features

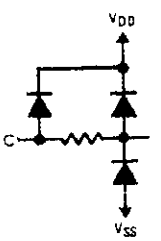
- Low power TTL compatibility — Fan out of 2 driving 74L or 1 driving 74LS
- 5V–10V–15V parametric ratings
- Symmetrical output characteristics
- Maximum input leakage 1 μA at 15V over full temperature range

## Schematic Diagrams




TL/F/5939–1

CD4001BC/BM



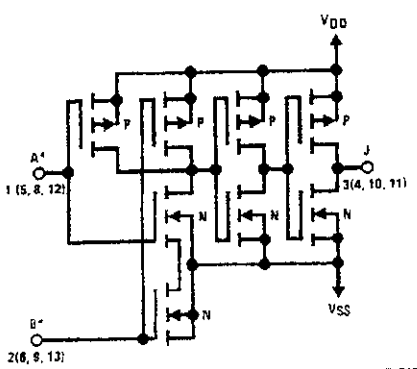

¼ of device shown

$J = \overline{A + B}$

Logical "1" = High
Logical "0" = Low

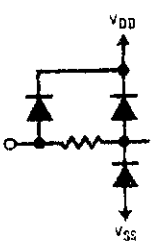
TL/F/5939–2

*All inputs protected by standard CMOS protection circuit




TL/F/5939–5

CD4011BC/BM




¼ of device shown

$J = \overline{A \cdot B}$

Logical "1" = High
Logical "0" = Low

TL/F/5939–8

*All inputs protected by standard CMOS protection circuit.



RRD-B30M105/Printed in U. S. A.

# XR-2206

Monolithic
Function Generator

EXAR ...the analog plus company™

June 1997-3

## FEATURES

- Low-Sine Wave Distortion, 0.5%, Typical
- Excellent Temperature Stability, 20ppm/°C, Typ.
- Wide Sweep Range, 2000:1, Typical
- Low-Supply Sensitivity, 0.01%V, Typ.
- Linear Amplitude Modulation
- TTL Compatible FSK Controls
- Wide Supply Range, 10V to 26V
- Adjustable Duty Cycle, 1% TO 99%

## APPLICATIONS

- Waveform Generation
- Sweep Generation
- AM/FM Generation
- V/F Conversion
- FSK Generation
- Phase-Locked Loops (VCO)

## GENERAL DESCRIPTION

The XR-2206 is a monolithic function generator integrated circuit capable of producing high quality sine, square, triangle, ramp, and pulse waveforms of high-stability and accuracy. The output waveforms can be both amplitude and frequency modulated by an external voltage. Frequency of operation can be selected externally over a range of 0.01Hz to more than 1MHz.

The circuit is ideally suited for communications, instrumentation, and function generator applications requiring sinusoidal tone, AM, FM, or FSK generation. It has a typical drift specification of 20ppm/°C. The oscillator frequency can be linearly swept over a 2000:1 frequency range with an external control voltage, while maintaining low distortion.

## ORDERING INFORMATION

| Part No. | Package | Operating Temperature Range |
|---|---|---|
| XR-2206M | 16 Lead 300 Mil CDIP | -55°C to +125°C |
| XR-2206P | 16 Lead 300 Mil PDIP | –40°C to +85°C |
| XR-2206CP | 16 Lead 300 Mil PDIP | 0°C to +70°C |
| XR-2206D | 16 Lead 300 Mil JEDEC SOIC | 0°C to +70°C |



EXAR Corporation, 48720 Kato Road, Fremont, CA 94538 ◆ (510) 668-7000 ◆ (510) 668-7017

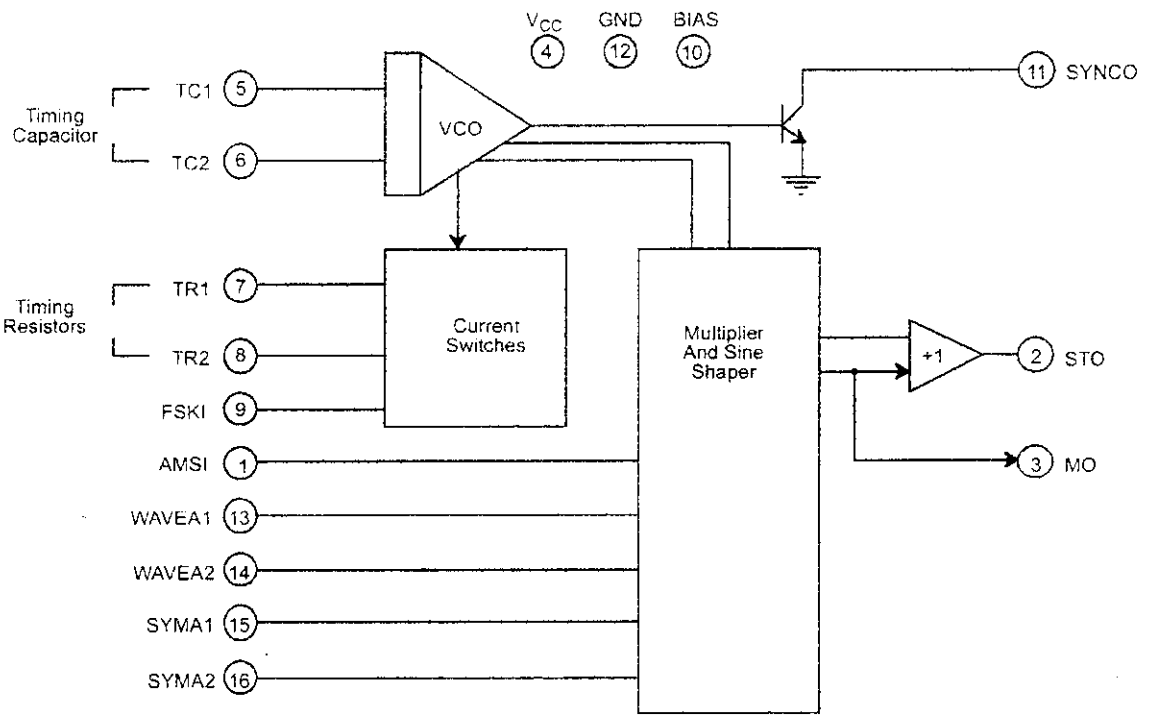
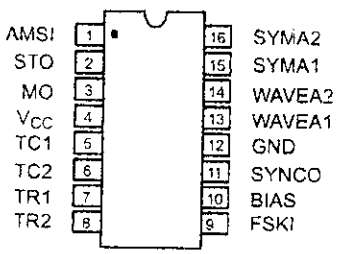
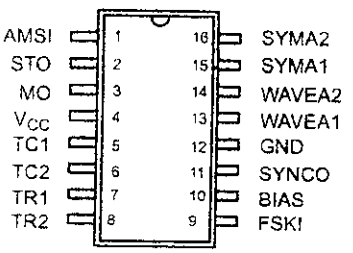



**Figure 1. XR-2206 Block Diagram**




16 Lead PDIP, CDIP (0.300")




16 Lead SOIC (Jedec, 0.300")

## PIN DESCRIPTION

| Pin # | Symbol | Type | Description |
|---|---|---|---|
| 1 | AMSI | I | **Amplitude Modulating Signal Input.** |
| 2 | STO | O | **Sine or Triangle Wave Output.** |
| 3 | MO | O | **Multiplier Output.** |
| 4 | $V_{CC}$ | | **Positive Power Supply.** |
| 5 | TC1 | I | **Timing Capacitor Input.** |
| 6 | TC2 | I | **Timing Capacitor Input.** |
| 7 | TR1 | O | **Timing Resistor 1 Output.** |
| 8 | TR2 | O | **Timing Resistor 2 Output.** |
| 9 | FSKI | I | **Frequency Shift Keying Input.** |
| 10 | BIAS | O | **Internal Voltage Reference.** |
| 11 | SYNCO | O | **Sync Output.** This output is a open collector and needs a pull up resistor tc $V_{CC}$. |
| 12 | GND | | **Ground pin.** |
| 13 | WAVEA1 | I | **Wave Form Adjust Input 1.** |
| 14 | WAVEA2 | I | **Wave Form Adjust Input 2.** |
| 15 | SYMA1 | I | **Wave Symetry Adjust 1.** |
| 16 | SYMA2 | I | **Wave Symetry Adjust 2.** |

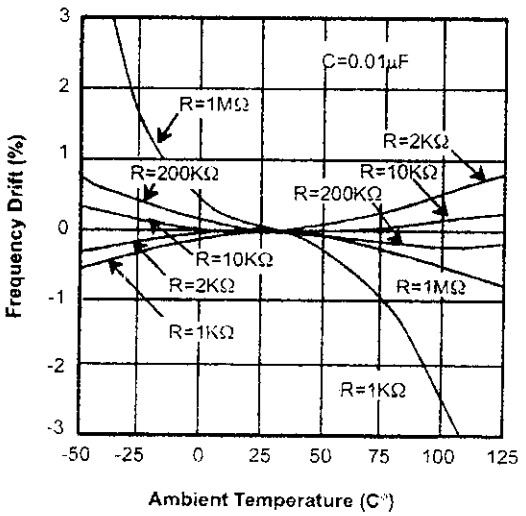



Figure 9. Frequency Drift versus Temperature.

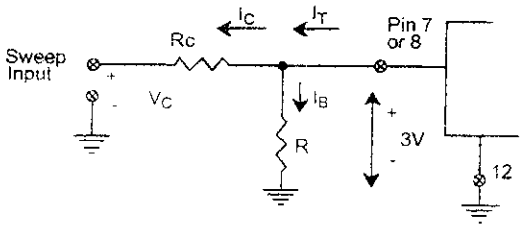



Figure 10. Circuit Connection for Frequency Sweep.

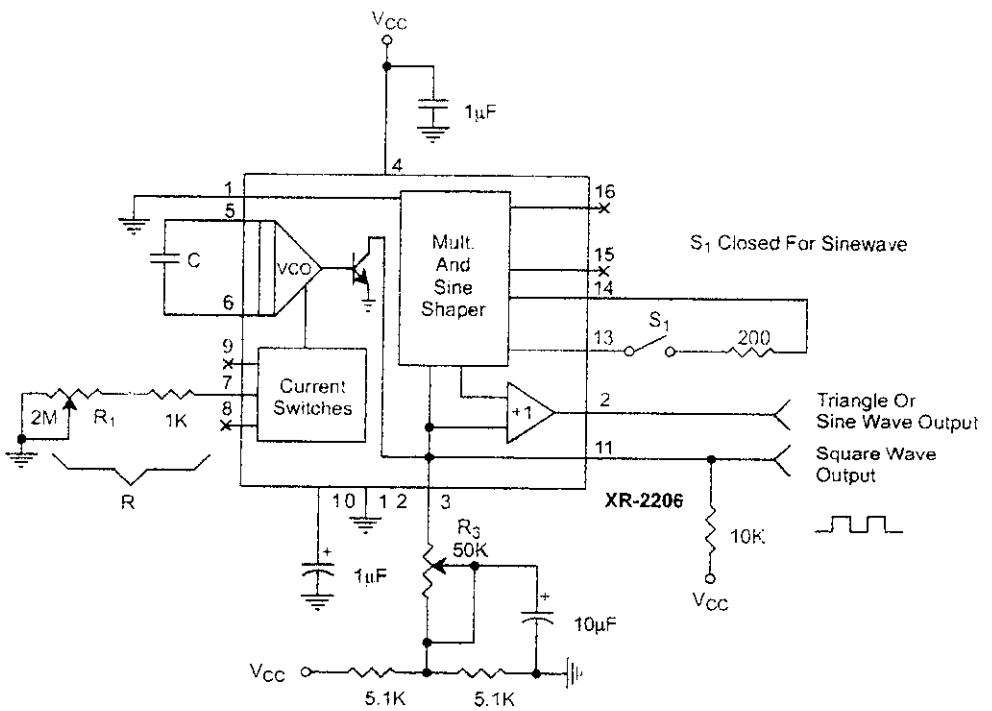



Figure 11. Circuit tor Sine Wave Generation without External Adjustment. (See *Figure 3* for Choice of $R_3$)

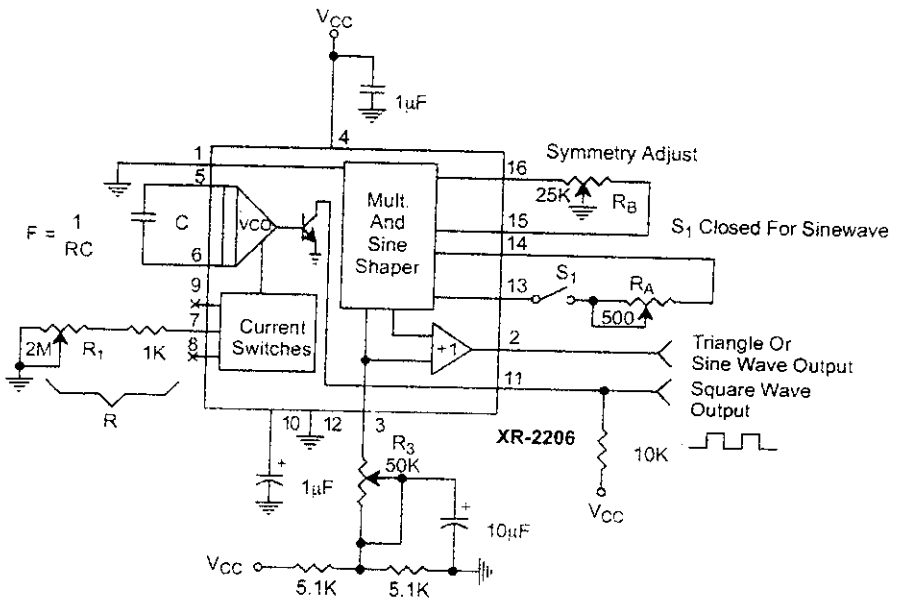



**Figure 12. Circuit for Sine Wave Generation with Minimum Harmonic Distortion. ($R_3$ Determines Output Swing - See *Figure 3*)**

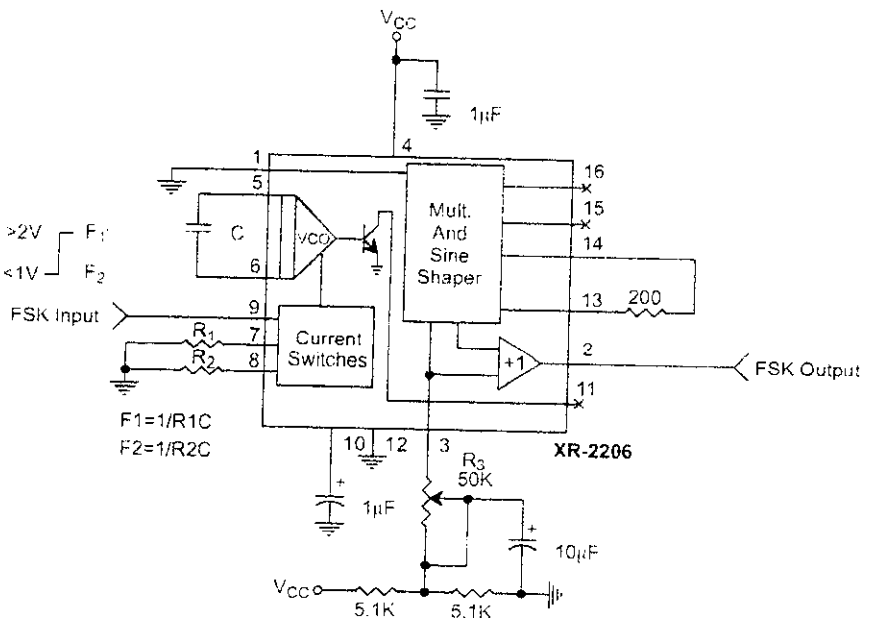



**Figure 13. Sinusoidal FSK Generator**

### With External Adjustment:

The harmonic content of sinusoidal output can be reduced to -0.5% by additional adjustments as shown in *Figure 12*. The potentiometer, $R_A$, adjusts the sine-shaping resistor, and $R_B$ provides the fine adjustment for the waveform symmetry. The adjustment procedure is as follows:

1. Set $R_B$ at midpoint and adjust $R_A$ for minimum distortion.
2. With $R_A$ set as above, adjust $R_B$ to further reduce distortion.

### Triangle Wave Generation

The circuits of *Figure 11* and *Figure 12* can be converted to triangle wave generation, by simply open-circuiting Pin 13 and 14 (i.e., $S_1$ open). Amplitude of the triangle is approximately twice the sine wave output.

### FSK Generation

*Figure 13* shows the circuit connection for sinusoidal FSK signal operation. Mark and space frequencies can be independently adjusted by the choice of timing resistors, $R_1$ and $R_2$; the output is phase-continuous during transitions. The keying signal is applied to Pin 9. The circuit can be converted to split-supply operation by simply replacing ground with $V^-$.

### Pulse and Ramp Generation

*Figure 14* shows the circuit for pulse and ramp waveform generation. In this mode of operation, the FSK keying terminal (Pin 9) is shorted to the square-wave output (Pin 11), and the circuit automatically frequency-shift keys itself between two separate frequencies during the positive-going and negative-going output waveforms. The pulse width and duty cycle can be adjusted from 1% to 99% by the choice of $R_1$ and $R_2$. The values of $R_1$ and $R_2$ should be in the range of 1kΩ to 2MΩ.

## PRINCIPLES OF OPERATION

### Description of Controls

### Frequency of Operation:

The frequency of oscillation, $f_o$, is determined by the external timing capacitor, C, across Pin 5 and 6, and by the timing resistor, R, connected to either Pin 7 or 8. The frequency is given as:

$$f_o = \frac{1}{RC}\ Hz$$

and can be adjusted by varying either R or C. The recommended values of R, for a given frequency range, as shown in *Figure 5*. Temperature stability is optimum for $4k\Omega < R < 200k\Omega$. Recommended values of C are from 1000pF to 100μF.

### Frequency Sweep and Modulation:

Frequency of oscillation is proportional to the total timing current, $I_T$, drawn from Pin 7 or 8:

$$f = \frac{320 I_T(mA)}{C(\mu F)}\ Hz$$

Timing terminals (Pin 7 or 8) are low-impedance points, and are internally biased at +3V, with respect to Pin 12. Frequency varies linearly with IT, over a wide range of current values, from 1μA to 3mA. The frequency can be controlled by applying a control voltage, $V_C$, to the activated timing pin as shown in *Figure 10*. The frequency of oscillation is related to VC as:

$$f = \frac{1}{RC}\left(1 + \frac{R}{R_c}\left(1 - \frac{V_c}{3}\right)\right) Hz$$

where $V_C$ is in volts. The voltage-to-frequency conversion gain, K, is given as:

$$K = \partial f / \partial V_c = -\frac{0.32}{R_c C}\ \ Hz/V$$

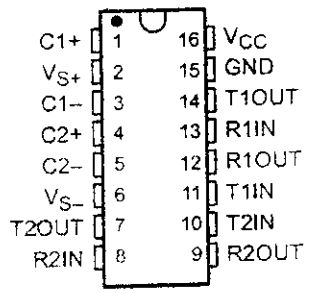
***CAUTION:*** *For safety operation of the circuit, $I_T$ should be limited to ≤ 3mA.*

# MAX232, MAX232I
# DUAL EIA-232 DRIVER/RECEIVER

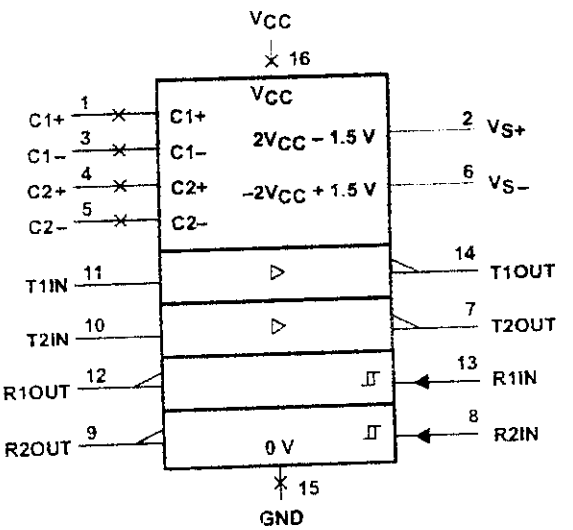
SLLS047G – FEBRUARY 1989 – REVISED AUGUST 1998

- Operates With Single 5-V Power Supply
- LinBiCMOS™ Process Technology
- Two Drivers and Two Receivers
- ±30-V Input Levels
- Low Supply Current . . . 8 mA Typical
- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Designed to be Interchangeable With Maxim MAX232
- Applications
  - TIA/EIA-232-F
  - Battery-Powered Systems
  - Terminals
  - Modems
  - Computers
- ESD Protection Exceeds 2000 V Per MIL-STD-883, Method 3015
- Package Options Include Plastic Small-Outline (D, DW) Packages and Standard Plastic (N) DIPs

D, DW, OR N PACKAGE
(TOP VIEW)




## logic symbol†




† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

## description

The MAX232 device is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

The MAX232 is characterized for operation from 0°C to 70°C. The MAX232I is characterized for operation from –40°C to 85°C.

**AVAILABLE OPTIONS**

| $T_A$ | PACKAGED DEVICES | | |
|---|---|---|---|
| | SMALL OUTLINE (D) | SMALL OUTLINE (DW) | PLASTIC DIP (N) |
| 0°C to 70°C | MAX232D‡ | MAX232DW‡ | MAX232N |
| –40°C to 85°C | MAX232ID‡ | MAX232IDW‡ | MAX232IN |

‡ This device is available taped and reeled by adding an R to the part number (i.e., MAX232DR).

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC and LinBiCMOS are trademarks of Texas Instruments Incorporated.





TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

### absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Input supply voltage range, $V_{CC}$ (see Note 1) . . . . . −0.3 V to 6 V
Positive output supply voltage range, $V_{S+}$ . . . . . $V_{CC}$ −0.3 V to 15 V
Negative output supply voltage range, $V_{S-}$ . . . . . −0.3 V to −15 V
Input voltage range, $V_I$: Driver . . . . . −0.3 V to $V_{CC}$ + 0.3 V
Receiver . . . . . ±30 V
Output voltage range, $V_O$: T1OUT, T2OUT . . . . . $V_{S-}$ −0.3 V to $V_{S+}$ + 0.3 V
R1OUT, R2OUT . . . . . −0.3 V to $V_{CC}$ + 0.3 V
Short-circuit duration: T1OUT, T2OUT . . . . . Unlimited
Package thermal impedance, $\theta_{JA}$ (see Note 2): D package . . . . . 113°C/W
DW package . . . . . 105°C/W
N package . . . . . 78°C/W
Storage temperature range, $T_{stg}$ . . . . . −65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds . . . . . 260°C

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

NOTE 1: All voltage values are with respect to network ground terminal.
2. The package thermal impedance is calculated in accordance with JESD 51, except for through-hole packages, which use a trace length of zero.

### recommended operating conditions

| | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|
| Supply voltage, $V_{CC}$ | | 4.5 | 5 | 5.5 | V |
| High-level input voltage, $V_{IH}$ (T1IN,T2IN) | | 2 | | | V |
| Low-level input voltage, $V_{IL}$ (T1IN, T2IN) | | | | 0.8 | V |
| Receiver input voltage, R1IN, R2IN | | | | ±30 | V |
| Operating free-air temperature, $T_A$ | MAX232 | 0 | | 70 | °C |
| | MAX232I | –40 | | 85 | |

## APPLICATION INFORMATION

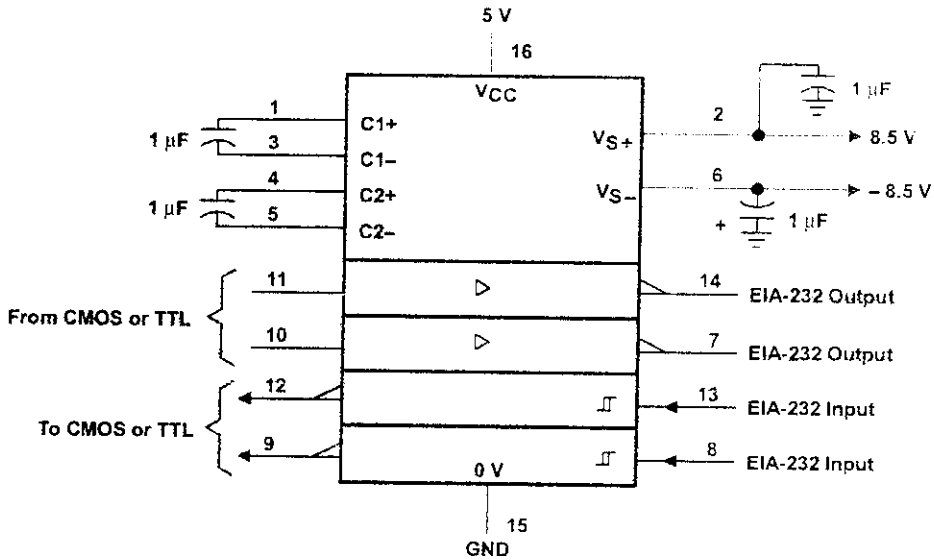



Figure 4. Typical Operating Circuit