REFERENCES

Website

[1] Author unknown, (1997), Stanford University Humming Bird project, http://sun-valley.stanford.edu/~heli/

[2] Paul DeBitetto (1995 to 1997), MIT University Draper project, http://web.mit.edu/whall/www/heli/

[3] Carnegie Mellon Robotics Institute, (1998). Carnegie Mellon Autonomous helicopter project, http://www.cs.cmu.edu/afs/cs/project/chopper/www/heli_project.html

[4] Wolfgang Paulus (1998), ECP-II Project, http:// wolfpaulus.com/bio/ucsd/cmoscam.pdf

[5] Digital Analog Component (2007) 8051 Webcam Application Note www.dacomwest.de/pdf/8051nc_application_note.pdf

[6] Wade Tyler Capenter, (2001). Wireless Roaming Webcam, http://www.ece.ubc.ca/~elec474/reports/dec01/group2.pdf

[7] James W. Stewart (1999) The 8051 Microcontroller Hardware, Software and Interfacing, second edition, pp. 32-60

Reference Books

[8] James W.Stewart (1999*). The 8051 Microcontroller*. 2nd Edition, Prentice-Hall. USA.

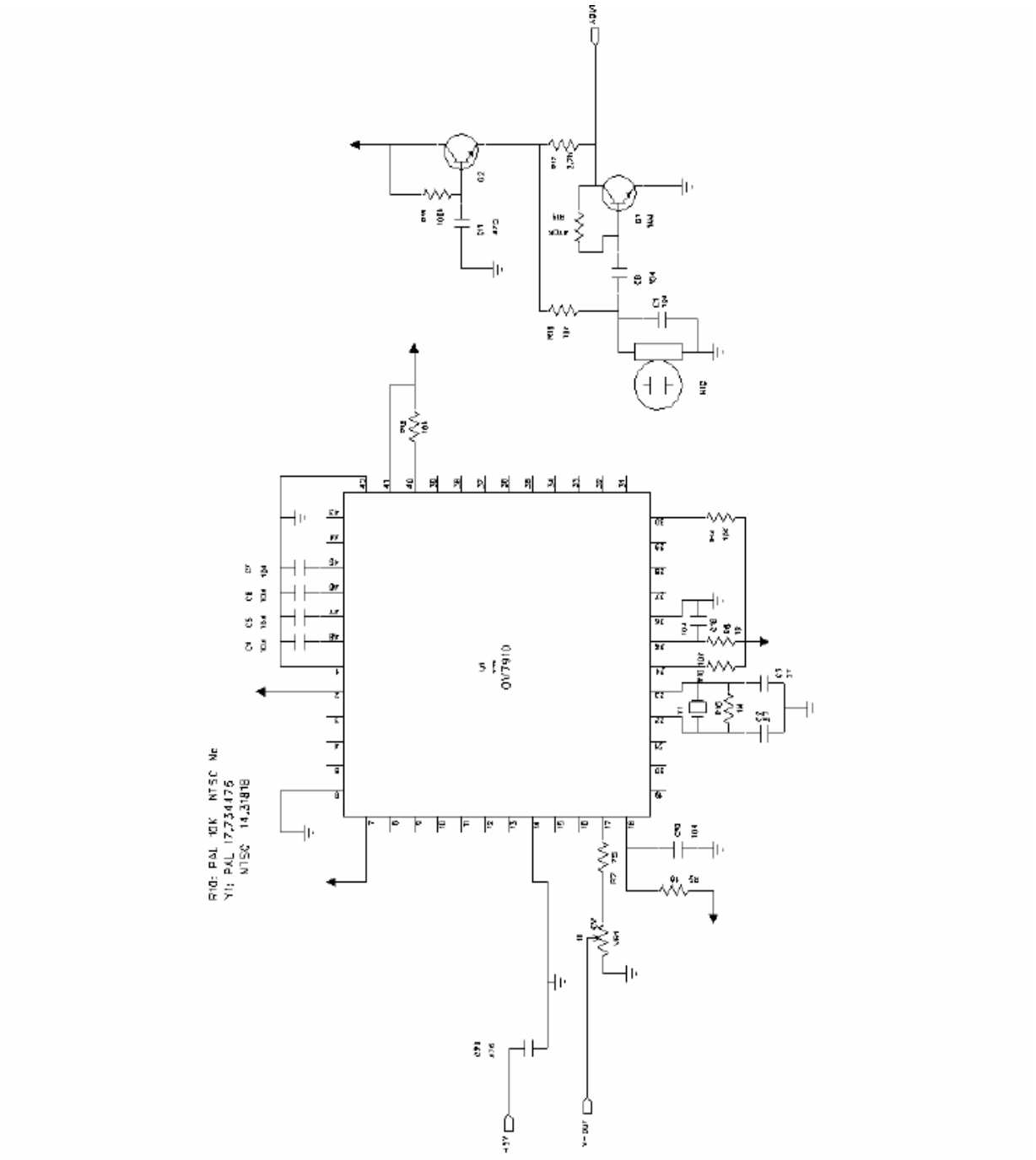
[9] Louis E.Frenzel (2000). *Communication Electronics.* 3rd Edition, Glencoe / McGraw-Hill. Singapore.

[10] Greg Perry with Sanjaya Hettihewa (1998). *Sams Teach Yourself Visual Basic® 6.* 1st Edition, Sams Publishing. USA.

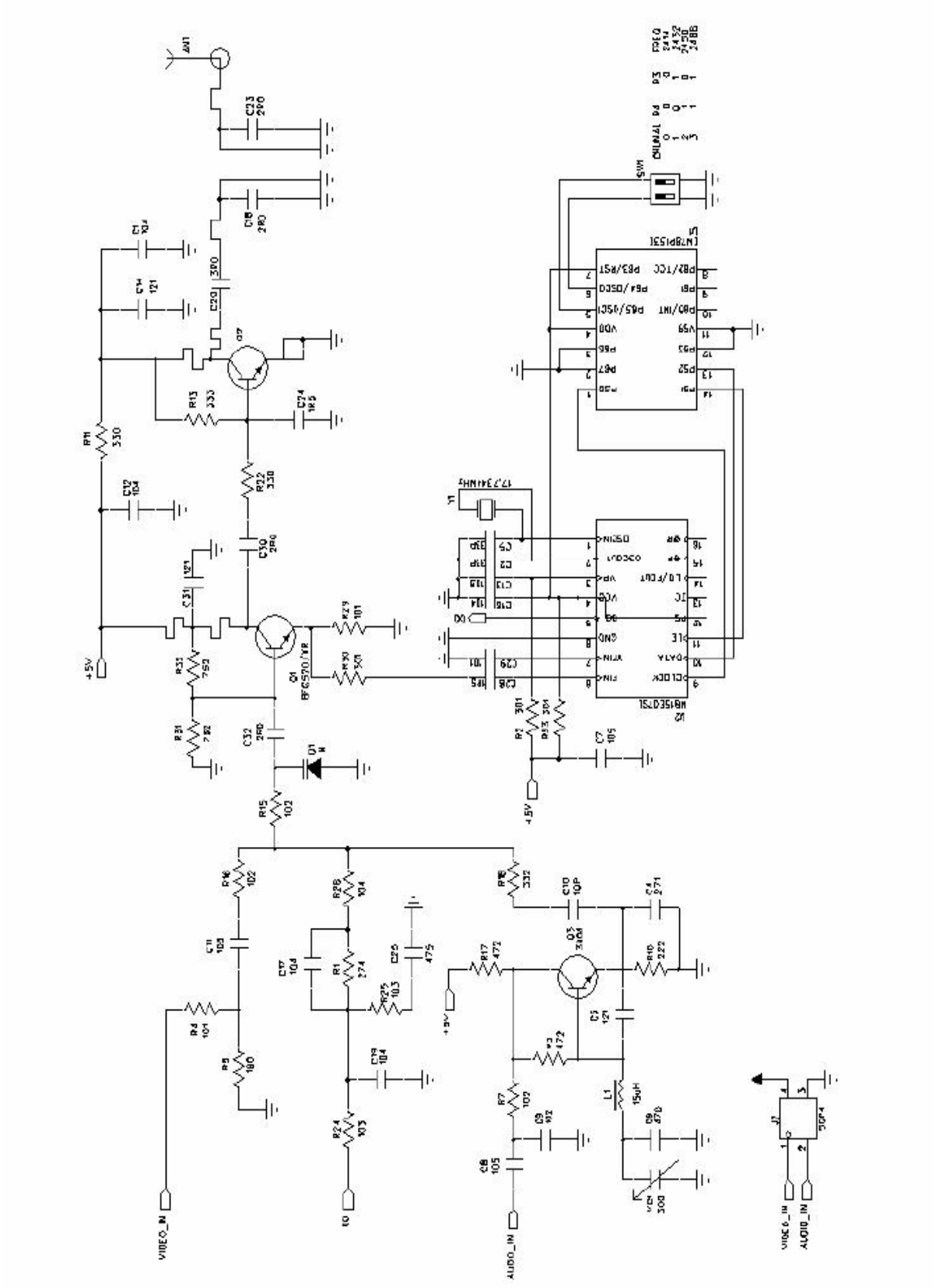
[11] Wayne Tomasi (2004). Electronic Communication Systems. 5th Edition Prentice-Hall. Singapore.

APPENDICES

Appendix A: Camera Schematic I



Appendix B: Camera Schematic II



Appendix C: User Interface Visual Basic Code

```
Private Sub Command2_Click()
If Command2.Enabled = True Then Timer1.Enabled = True
If Command2.Enabled = True Then Picture1.Picture = Clipboard.GetData:
Clipboard.Clear
If Command2.Enabled = True Then Picture1.Picture =
LoadPicture("incomingsignal.bmp")
End Sub
```

```
Private Sub Command3_Click()
If Command3.Enabled = True Then Timer1.Enabled = False
If Command3.Enabled = True Then Picture1.Picture = Clipboard.GetData:
Clipboard.Clear
If Command3.Enabled = True Then Picture1.Picture =
LoadPicture("signalpaused.bmp")
End Sub
```

```
Private Sub Command4_Click()
Text1.Text = ""
txtBinary.Text = ""
'Text1.Text = LoadPicture("nosignal.bmp")
If Command4.Enabled = True Then Picture1.Picture = Clipboard.GetData:
Clipboard.Clear
If Command4.Enabled = True Then Picture1.Picture = LoadPicture("nosignal.bmp")
End Sub
```

```
Private Sub Form_Load()
MSComm1.CommPort = 1
MSComm1.Settings = "1200,N,8,1"
MSComm1.InputLen = 0
MSComm1.PortOpen = True
```

```
Timer2.Enabled = True
Timer1.Enabled = False
Picture1.Picture = LoadPicture("nosignal.bmp")
'Set Skinner1.Forms = Forms
'If Command2.Enabled = False Then Timer1.Enabled = False
End Sub
```

```
Private Sub Timer1_Timer()
Dim txt As String
Dim result As String
Dim ch As String
Dim bin As String
Dim i As Integer
Dim DISPLAY As String
DISPLAY = MSComm1.Input
Text1.Text = DISPLAY
MSComm1.CDTimeout = 5000

Call save_file
Call save_file1
txt = Text1.Text
txt = Replace(txt, vbCr, "")
txt = Replace(txt, vbLf, "")

result = ""
For i = 1 To Len(txt)
ch = Mid$(txt, i, 1)

bin = LongToBinary(Asc(ch), False)
result = result & Right$(bin, 8)
Next i
```

```
txtBinary.Text = result
End Sub
```

```
Private Sub Timer2_Timer()
Label4.Caption = Format(Now, "medium date")
Label5.Caption = Format(Now, "long time")
End Sub
```

```
Private Function LongToBinary(ByVal long_value As Long, Optional ByVal
separate_bytes As Boolean = True) As String
Dim hex_string As String
Dim digit_num As Integer
Dim digit_value As Integer
Dim nibble_string As String
Dim result_string As String
Dim factor As Integer
Dim bit As Integer

' Convert into hex.
hex_string = Hex$(long_value)

' Zero-pad to a full 8 characters.
hex_string = Right$(String$(8, "0") & hex_string, 8)

' Read the hexadecimal digits
' one at a time from right to left.
For digit_num = 8 To 1 Step -1
' Convert this hexadecimal digit into a
' binary nibble.
digit_value = CLng("&H" & Mid$(hex_string, _
digit_num, 1))
```

```
' Convert the value into bits.
factor = 1
nibble_string = ""
For bit = 3 To 0 Step -1
If digit_value And factor Then
nibble_string = "1" & nibble_string
Else
nibble_string = "0" & nibble_string
End If
factor = factor * 2
Next bit

' Add the nibble's string to the left of the
' result string.
result_string = nibble_string & result_string
Next digit_num

' Add spaces between bytes if desired.
If separate_bytes Then
result_string = _
Mid$(result_string, 1, 8) & " " & _
Mid$(result_string, 9, 8) & " " & _
Mid$(result_string, 17, 8) & " " & _
Mid$(result_string, 25, 8)
End If

' Return the result.
LongToBinary = result_string
End Function
```

```
Private Sub save_file()
saving = FreeFile
Open "c:\data.txt" For Append As saving
Print #saving, Text1.Text
Close saving
End Sub
```

```
Private Sub save_file1()
saving = FreeFile
Open "c:\Binarydata.txt" For Append As saving
Print #saving, txtBinary.Text
Close saving

End Sub
```

Appendix D: Microcontroller Program

```
        ORG     0000H

        MOV     P0, #0FFH
        MOV     P2, #0FFH
        MOV     P1, #0FFH

    ;----------------------------------------
    ;SET BOUD RATE:

        MOV     TMOD, #20H      ;TIMER1, MODE2
        MOV     TH1, #0E8H ;(E8H  1200)     ;300 BAUD RATE 0A0H
        MOV     SCON, #50H      ;USE SERIAL MODE 1 ,ENABLE RX
        SETB    TR1

    MAIN:
        CLR     P1.0
    CMORE:  JNB     P1.1, RUN
        JNB     P1.2, CLESS                     ;Check bit
        JNB     P1.3, CLESS
        JNB     P1.4, CLESS
        JNB     P1.5, CLESS
        CLR     P1.5
        JNB     P1.6, CLESS
        JNB     P1.7, CLESS
        CLR     P1.7
    CLESS:      JNB     P1.1, RUN
        JB      P1.2, RUN
        SETB P1.2
```

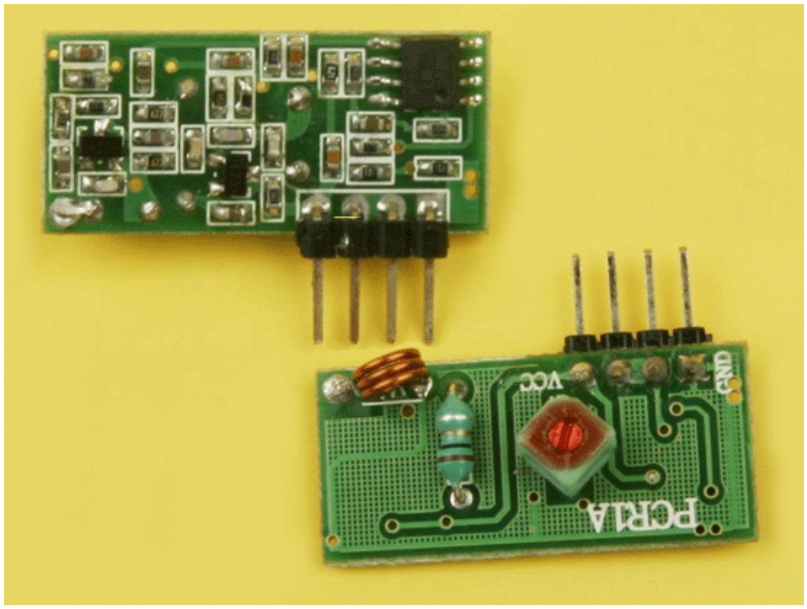
```
RUN:        MOV   A, P1          ;#021H
     CALL  SEND
     JMP     MAIN


;----------------------------------------
;SERIAL DATA SENDING SUB

SEND:
     MOV     SBUF, A
WAIT:
     JNB     TI, WAIT
     CLR     TI
     call      DLY
     RET

;----------------------------------------
DLY:
     MOV     R1, #20
D1:     MOV     R2, #20
D2:     NOP
     NOP
     DJNZ    R2, D2
     DJNZ    R1, D1
     RET
DLY1:
     MOV     R1, #4
D3:     MOV     R2, #250
D4:          MOV   R3, #250
D5:     NOP
     NOP
```

```
DJNZ    R3, D5
DJNZ    R2, D4
DJNZ  R1, D3
RET
END
```

Appendix E: ASCII code

0008=<backspac>
0009 = <tab>
0010 = <down>
0013 = <down>
0022 = <paste >
0032 = <space>
0033 = !
0034 = "
0035 = #
0036 = $
0037 = %
0038 = &
0039 = '
0040 = (
0041 =)
0042 = *
0043 = +
0044 = ,
0045 = -
0046 = .
0047 = /
0048 = 0
0049 = 1
0050 = 2
0051 = 3
0052 = 4
0053 = 5
0054 = 6
0055 = 7
0057 = 8
0058 = 9
0059 = ;
0060 = <
0061 = =
0062 = >
0063 = ?
0064 = @
0065 = A
0066 = B
0067 = C
0068 = D
0069 = E
0070 = F
0071 = G
0072 = H
0073 = I
0074 = J
0075 = K
0076 = L
0077 = M
0078 = N
0079 = O
0080 = P
0081 = Q
0082 = R
0083 = S
0084 = T
0085 = U
0086 = V
0087 = W
0088 = X
0089 = Y
0090 = Z
0091 = [
0092 = \
0093 =]
0094 = ^
0095 = _
0096 = `
0097 = a
0098 = b
0099 = c
0100 = d
0101 = e
0102 = f
0103 = g
0104 = h
0105 = i
0106 = j
0107 = k
0108 = l
0109 = m
0110 = n
0111 = o
0112 = p
0113 = q
0114 = r
0115 = s
0116 = t
0117 = u
0118 = v
0119 = w
0120 = x
0121 = y
0122 = z
0123 = {
0124 = |
0125 = }
0126 = ~
0127 = □
0128 = €
0129 = □
0130 = ,
0131 = ƒ
0132 = „
0133 = …
0134 = †
0135 = ‡
0136 = ˆ
0137 = ‰
0138 = Š
0139 = ‹
0140 = Œ
0141 = □
0142 = Ž
0143 = □
0144 = □
0145 = ‘
0146 = ’
0147 = "
0148 = "
0149 = •
0150 = –
0151 = —
0152 = ˜
0153 = ™
0154 = š
0155 = ›
0156 = œ
0157 = □
0158 = ž
0159 = Ÿ
0160 = <space>
0161 = ¡
0162 = ¢
0163 = £
0164 = ¤
0165 = ¥
0166 = ¦
0167 = §
0168 = ¨
0169 = ©
0170 = ª
0171 = «
0172 = ¬
0173 =
0174 = ®
0175 = ¯
0176 = °
0177 = ±
0178 = ²
0179 = ³
0180 = ´
0181 = µ
0182 = ¶
0183 = ·
0184 = ¸
0185 = ¹
0186 = º
0187 = »
0188 = ¼
0189 = ½
0190 = ¾
0191 = ¿
0192 = À
0193 = Á
0194 = Â
0195 = Ã
0196 = Ä
0197 = Å
0198 = Æ
0199 = Ç
0200 = È
0201 = É
0202 = Ê
0203 = Ë
0204 = Ì
0205 = Í
0206 = Î
0207 = Ï
0208 = Ð
0209 = Ñ
0210 = Ò
0211 = Ó
0212 = Ô
0213 = Õ
0214 = Ö
0215 = ×
0216 = Ø
0217 = Ù
0218 = Ú
0219 = Û
0220 = Ü
0221 = Ý
0222 = Þ
0223 = ß
0224 = à
0225 = á
0226 = â
0227 = ã
0228 = ä
0229 = å
0230 = æ
0231 = ç
0232 = è
0233 = é
0234 = ê
0235 = ë
0236 = ì
0237 = í
0238 = î
0239 = ï
0240 = ð
0241 = ñ
0242 = ò
0243 = ó
0244 = ô
0245 = õ
0246 = ö
0247 = ÷
0248 = ø
0249 = ù
0250 = ú
0251 = û
0252 = ü
0253 = ý
0254 = þ
0255 = ÿ

Appendix F: PCR1A-specificaiton



Nominal DC voltage : (5V+0.5V)

Current work: 4mA@5V

Frequency: 315MHZ/433MHZ

Output data: TTL level

Num of Pin: 4PIN

Maximum speed: 1KHZ

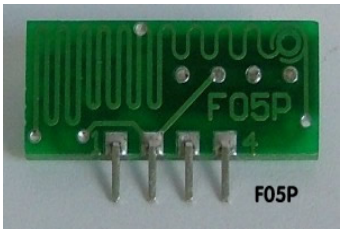
Receiver sensitivity: 105 DBm

Maximum size: 47 x 15 x 12mm appearance

Temperature: -10 degrees ~+60 ℃

Characteristics: low voltage and high performance-price ratio, otherwise 1.7mA@5V Small Current high sensitivity receiver

Appendix G: F05P Specification



Transmitting frequency 315MHz or 433MHz,

Voltage: 3-12V

Emission current: 2-10mA

Fired power:10mW,

Frequency stability: 10^{-5} (Table stable frequency acoustic),

Operating temperature range: +60 ° C to -40 ° C

Size :9 × 21 × 5 mm (width × high thick)

The allocation of appropriate MCU

Appendix H: ATMEL 89S52 Datasheet

Features

- Compatible with MCS-51® Products
- 8K Bytes of In-System Programmable (ISP) Flash Memory
  - Endurance: 1000 Write/Erase Cycles
- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Eight Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- Power-off Flag

Description

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 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 in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 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 interrupt or hardware reset.

ATMEL®

8-bit Microcontroller with 8K Bytes In-System Programmable Flash

AT89S52

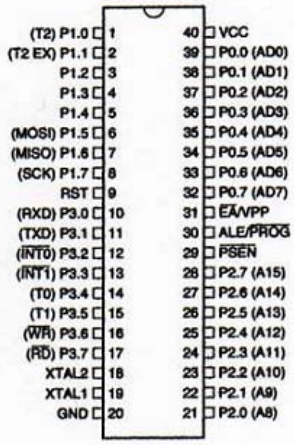
Preliminary

Rev. 1919A-07/01

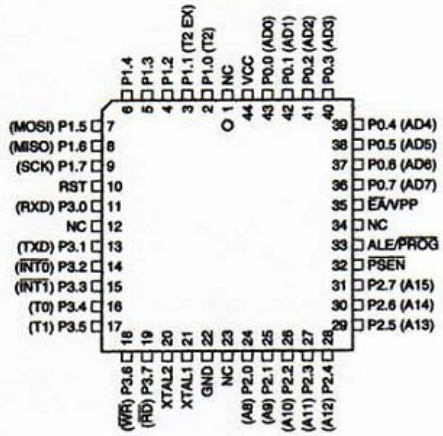
ATMEL

Pin Configurations

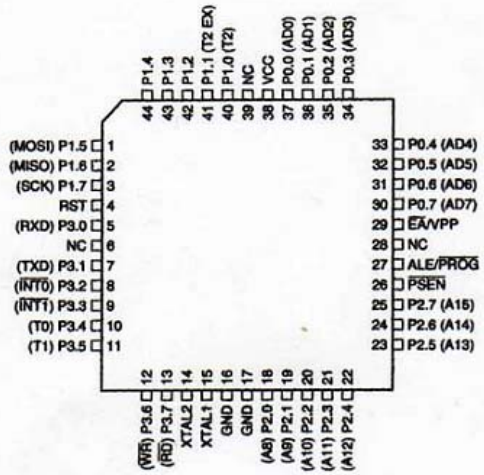
PDIP



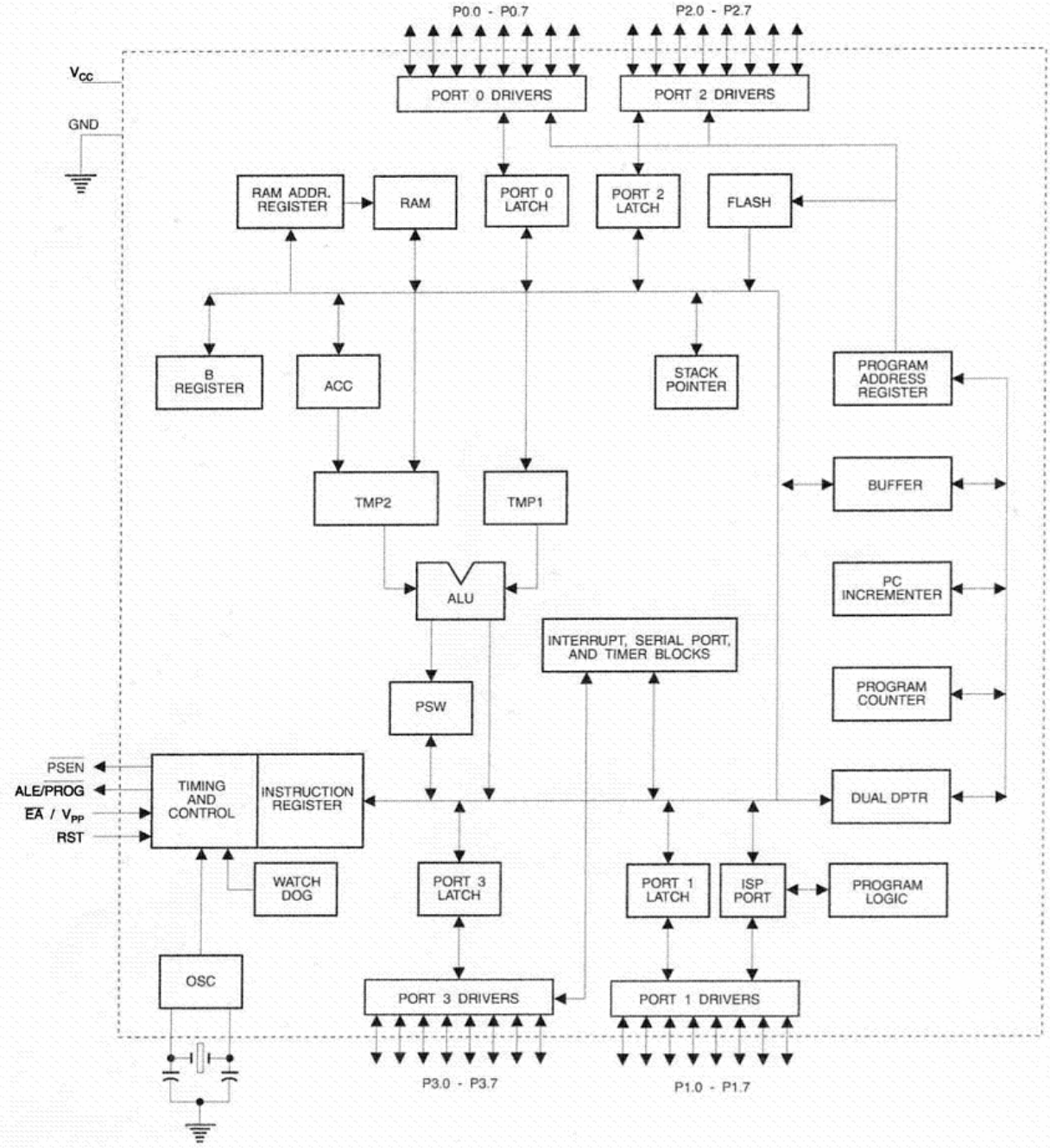
PLCC



TQFP



Block Diagram



Pin Description

VCC
Supply voltage.

GND
Ground.

Port 0
Port 0 is an 8-bit open drain bidirectional 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 can 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 bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.

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

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

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

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

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

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

Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.

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

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)

RST
Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 96 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

ALE/$\overline{PROG}$
Address Latch Enable (ALE) is an 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 ($\overline{\text{PSEN}}$) is the read strobe to external program memory.

When the AT89S52 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.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Table 1. AT89S52 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000								0AFH
0A0H	P2 11111111		AUXR1 XXXXXXX0				WDTRST XXXXXXXX		0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0		8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XXX0000	87H

APPENDIX I: ADC0804 Datasheet

National Semiconductor

December 1994

ADC0801/ADC0802/ADC0803/ADC0804/ADC0805 8-Bit μP Compatible A/D Converters

General Description

The ADC0801, ADC0802, ADC0803, ADC0804 and ADC0805 are CMOS 8-bit successive approximation A/D converters that use a differential potentiometric ladder—similar to the 256R products. These converters are designed to allow operation with the NSC800 and INS8080A derivative control bus with TRI-STATE® output latches directly driving the data bus. These A/Ds appear like memory locations or I/O ports to the microprocessor and no interfacing logic is needed.

Differential analog voltage inputs allow increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

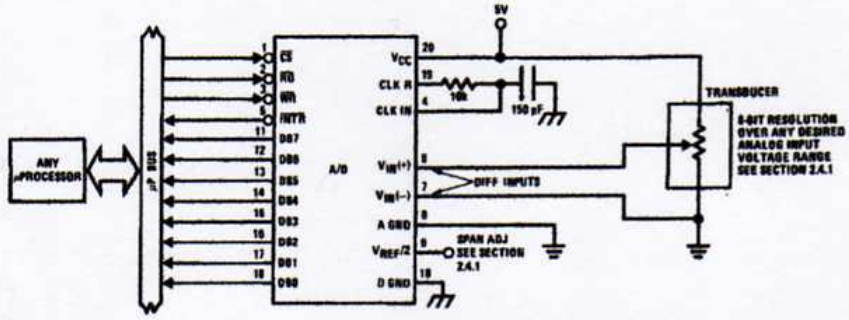
Features

- Compatible with 8080 μP derivatives—no interfacing logic needed - access time - 135 ns
- Easy interface to all microprocessors, or operates "stand alone"
- Differential analog voltage inputs
- Logic inputs and outputs meet both MOS and TTL voltage level specifications
- Works with 2.5V (LM336) voltage reference
- On-chip clock generator
- 0V to 5V analog input voltage range with single 5V supply
- No zero adjust required
- 0.3" standard width 20-pin DIP package
- 20-pin molded chip carrier or small outline package
- Operates ratiometrically or with 5 V_{DC}, 2.5 V_{DC}, or analog span adjusted voltage reference

Key Specifications

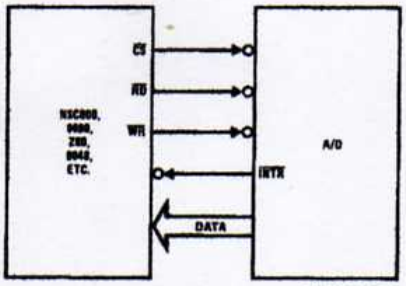
- Resolution: 8 bits
- Total error: ±¼ LSB, ±½ LSB and ±1 LSB
- Conversion time: 100 μs

Typical Applications



TL/H/5671–1

8080 Interface



TL/H/5671–31

Error Specification (Includes Full-Scale, Zero Error, and Non-Linearity)			
Part Number	Full-Scale Adjusted	$V_{REF}/2 = 2.500\ V_{DC}$ (No Adjustments)	$V_{REF}/2$ = No Connection (No Adjustments)
ADC0801	±¼ LSB		
ADC0802		±½ LSB	
ADC0803	±½ LSB		
ADC0804		±1 LSB	
ADC0805			±1 LSB

TRI-STATE® is a registered trademark of National Semiconductor Corp.
Z-80® is a registered trademark of Zilog Corp.

 TL/H/5671

RRD-B30M115/Printed in U. S. A.

ADC0801/ADC0802/ADC0803/ADC0804/ADC0805 8-Bit μP Compatible A/D Converters

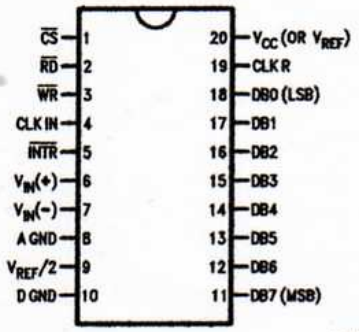
Ordering Information

TEMP RANGE		0°C TO 70°C	0°C TO 70°C	0°C TO 70°C	−40°C TO +85°C
ERROR	±¼ Bit Adjusted				ADC0801LCN
	±½ Bit Unadjusted	ADC0802LCWM	ADC0802LCV		ADC0802LCN
	±½ Bit Adjusted	ADC0803LCWM	ADC0803LCV		ADC0803LCN
	±1Bit Unadjusted	ADC0804LCWM	ADC0804LCV	ADC0804LCN	ADC0805LCN
PACKAGE OUTLINE		M20B—Small Outline	V20A—Chip Carrier	N20A—Molded DIP	

TEMP RANGE		−40°C TO +85°C	−55°C TO +125°C
ERROR	±¼ Bit Adjusted	ADC0801LCJ	ADC0801LJ
	±½ Bit Unadjusted	ADC0802LCJ	ADC0802LJ,
	±½ Bit Adjusted	ADC0803LCJ	ADC0802LJ/883
	±1Bit Unadjusted	ADC0804LCJ	
PACKAGE OUTLINE		J20A—Cavity DIP	J20A—Cavity DIP

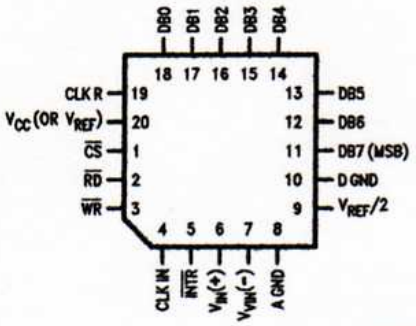
Connection Diagrams

ADC080X
Dual-In-Line and Small Outline (SO) Packages



TL/H/5671–30

ADC080X
Molded Chip Carrier (PCC) Package

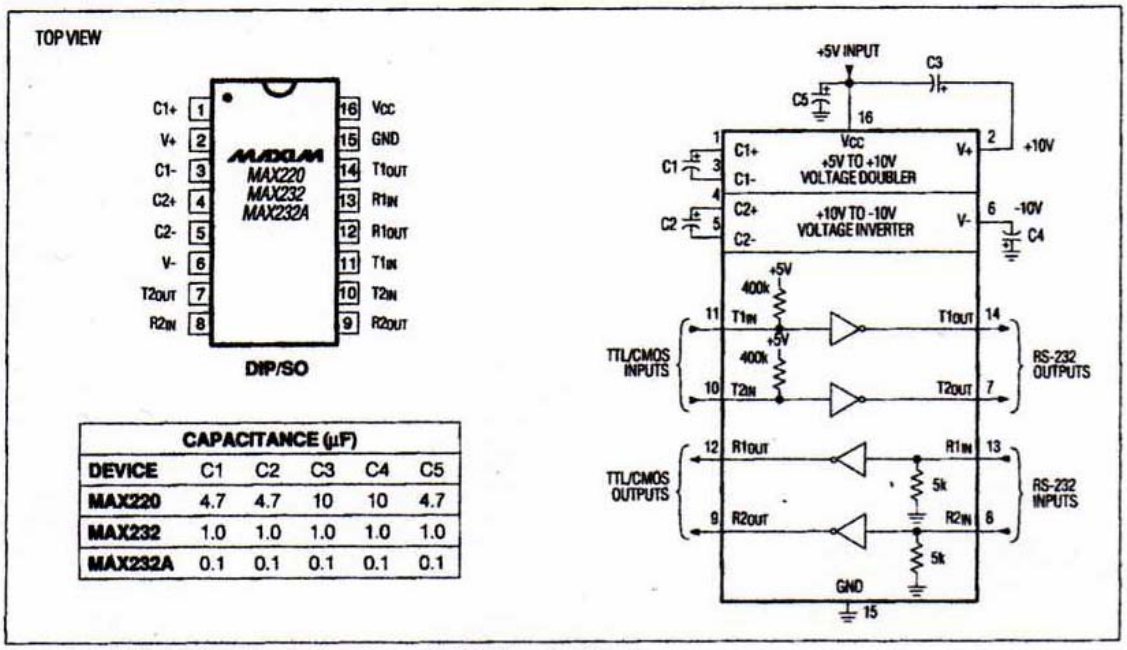


TL/H/5671–32

See Ordering Information

Appendix J: MAX 232 Datasheet

+5V-Powered, Multi-Channel RS-232 Drivers/Receivers



CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

Figure 5. MAX220/232/232A Pin Configuration and Typical Operating Circuit

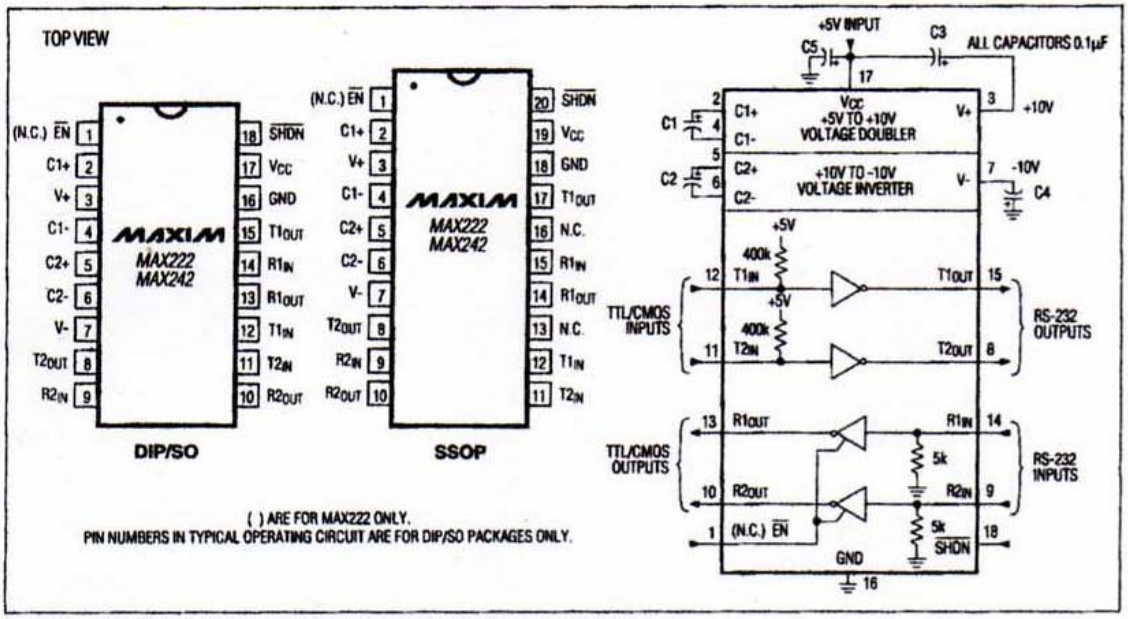


Figure 6. MAX222/MAX242 Pin Configuration and Typical Operating Circuit

MAXIM

Appendix K: Expenditure

Quantity	Description	Price
1	Microcontroller + Board	Rm 80.00
1	315 MHz RF module	Rm 85.00
1	Max232 board	Rm 10.00
1	Batery 9V	Rm 5.00
1	USB cable	Rm 4.00
1	RS 232 Connector	Rm 5.00
1	ADC 0804	Rm18.00
1	PCB	Rm 5.00
1	Connector	Rm6.00
4	Capacitor	Rm 1.60
1	Multi connector	Rm 1.20
	Total:	Rm 220.80
	Additional Cost:	Rm 15.00
	Grand Total:	Rm 235.80