Microcontroller based Washing Machine Control Through PC

PROJECT REPORT 2001 - 2002

Submitted by:

ANEESH K. MUKUNDAN

ANNAPOORANI V. R.

NITHYA R.

VIDYA P.

Under the guidance of

Prof. K. RAMPRAKASH, M.E.

2001 - 2002

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING IN ELECTRONICS AND COMMUNICATION ENGINEERING

of the Bharathiar University, Coimbatore.

Department of Electronics and Communication Engineering

Kumaraguru College of Technology

Coimbatore - 641006

Kumaraguru College of Technology

Coimbatore – 641006

Department of Electronics and Communication Engineering

Certificate

This is to certify that this project entitled

MICROCONTROLLER BASED WASHING MACHINE

CONTROL THROUGH PC

has been submitted by

ANEESH K. MUKUNDAN	**9827D0185**
ANNAPOORANI V.R	**9827D0186**
NITHYA R.	**9827D0208**
VIDYA P.	**9827D0240**

In partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in the Electronics and Communication Engineering Branch of the Bhatrath University, Coimbatore – 641006 during the academic year 2001–02

(Guide) 12/3/02

(Head of Department)

Certified that the candidates were examined by us in the Project Work.

Viva – Voce Examination held on 19 · 03 · 2002

(Internal Examiner)

(External Examiner)

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

We express our thanks and profound gratitude to our beloved Principal **Dr. K. K. Padmanabhan, B.Sc(Engg.)., M.E., Ph.D.,** for the constant support and encouragement he has given us to pursue new goals and ideas.

We are highly indebted to our beloved Professor and Head of the Department of Electronics and Communication Engineering **Prof. M. Ramasamy, M.E., M.I.S.T.E., M.I.E.E.E., M.I.E., C.(Engg.)., M.B.M.E.S.I., (Ph.D)** for his timely advice and remarkable guidance .

We are indeed privileged to have been under the guidance of **Prof. K. Ramprakash, M.E..** We thank him for the constant encouragement and moral support he provided for the completion of this project.

We express our heartfelt gratitude to **Mr. P. Manoj Kumar,** R & D Engineer, T. K. Controals for his immense help and thought provoking ideas without which the project would not have reached its stage. Also we thank **Ms. M. Gayathri,** T.K. Controals for her valuable guidance.

Our sincere thanks to our department faculty members for their inspiring advice and wholehearted support throughout this project.

Last but not the least, we extend our special thanks to our beloved parents and friends for their love and support during the development of this project.

CONTENTS

CONTENTS

SYNOPSIS

SYNOPSIS

Our project, "WASHING MACHINE CONTROL THROUGH PC" is a Microcontroller based project designed for automatic control of washing machine used in hospitals.

In our project, the microcontroller AT89C51 is chosen considering its several powerful features and programming flexibility. The RS485 communication used here enables several machines to be interfaced to a single PC. The required timings and temperature are set in the PC and this is programmed using VB 6.0. Sensors are used to sense the temperature and water level. The sensed temperature is converted to a digital equivalent and fed to the microcontroller.

The process starts with the water solenoid turning on, once the required water level is sensed. The controller maintains the set temperature by turning ON or OFF the steam solenoid as required. The forward and reverse rotations are controlled, as per the timing set in the PC, by the respective solenoids. Finally, when the total time elapses, the drain solenoid turns ON thereby stopping the process.

INTRODUCTION

1. Introduction

It's hard to imagine a more idyllic world without automation. Looking onto the bright side of technology you will find automation.

New world of information always demands a new kind of automation. Over the next few years systems will grow more capable and more widely available until they become standard equipment in luxury. This will raise the active digital technology to the next level enabling all home appliances to communicate and coordinate responses without a third man.

Automation can do a great deal to improve safety with present digital technology. There are already high end automatic systems/machines coupled to in the market and ours is one among those crowds.

Yes, our microcontroller based project "WASHING MACHINE CONTROL THROUGH PC" is a fully automatic system controlled by a PC through the RS485 communication.

2. PROJECT OVERVIEW

This project is basically a microcontroller based project which uses the AT89C51 microcontroller. The major criteria for choosing this project are many. Some of them are the reduced cost (since individual panel control is not required), easy error detection (since a single PC monitors all the machines) and maintaining a constant temperature throughout the process.

Using software, a front panel consisting of the buttons for the total time, forward time, reverse time, idle time, water solenoid, soap solenoid, steam solenoid, drain solenoid and the start, stop and exit buttons are created in PC for control of every machine. The timings and temperature as required by the user are entered into the text boxes created.

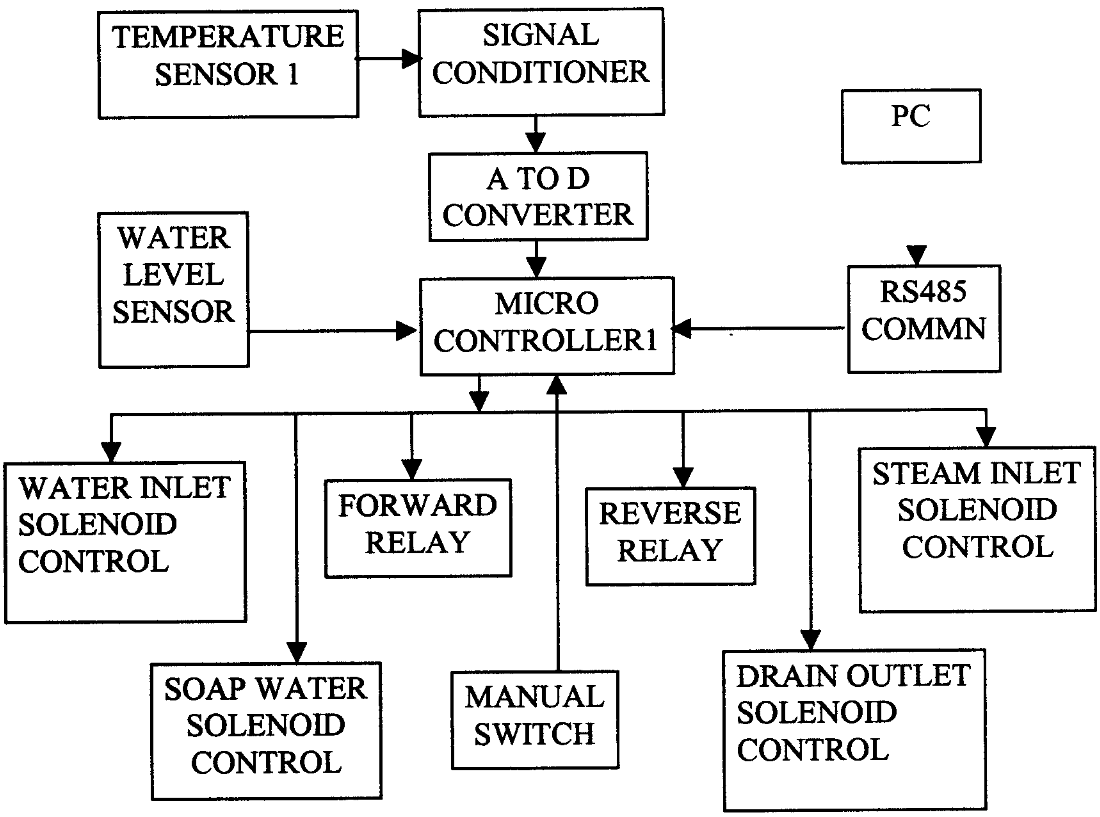
When the start button of the desired machine is pressed, the water solenoid turns ON and the water is gradually filled in the tank. When the water level reaches 1/3rd of the tank, the water sensor senses the water level and sends this signal to microcontroller to turn OFF the water solenoid. As soon as the water solenoid turns OFF, the soap solenoid automatically turns ON. After required soap is let inside the tank, a switch is manually pressed to turn off the soap solenoid.

Once the soap solenoid turns OFF, the motor starts rotating in forward direction, and when the forward time set by the user is reached the forward solenoid turns OFF and the motor remains idle for the set time. The motor then starts rotating in the reverse direction for the set time when the reverse solenoid turns ON. This is immediately followed by the idle position of the motor. This cycle of forward, idle, reverse, idle repeats itself till the total time elapses.

The temperature within the machine must be maintained constantly throughout the process. For this the water temperature is continuously compared with the set temperature. If the set temperature is greater than the current water temperature, the steam solenoid automatically turns ON and the temperature within the tank increases. Once the water temperature reaches the set temperature, the steam solenoid automatically turns OFF.

Once the total time is over, the drain outlet solenoid turns ON to let all the water outside. Thus the process comes to an end.

2.1 BLOCK DIAGRAM



HARDWARE DESCRIPTION

3. Hardware Description

The electronic devices and integrated chips are available for various purposes. But our aim is to connect them in a proper manner so that the system yields the desired result. In this modern developing world, the present trend is to relieve the burden of mankind by developing fully automatic machines.

Our project "WASHING MACHINE CONTROL THROUGH PC" which, as previously mentioned, is aimed at creating a fully automatic machine and capable of maintaining a constant set temperature within it throughout its entire process. The major highlight of this project is the RS485 communication which enables several machines (up to 32 machines) to be simultaneously controlled by a single PC (master and slave configuration). But for simplicity, in this project only 2 machines are interfaced to the PC.

This project is technically a printed circuit board with various ICs, sensors and other electronic component connected on it, powered by a supply and interfaced to the PC. But functionally, there is a lot more within it which is described in the following sections.

The hardware part of this project consists of the following parts.

- Temperature sensing
- Analog to digital converter ADC0809
- Microcontroller Atmel AT89C51
- RS485 serial interface

3.1 Temperature sensing

The most important criteria of this machine , as previously mentioned, is to maintain the temperature of the water within the machine. For this the microcontroller continuously compares the water temperature with the required temperature as set in the front panel of the PC. This is done using a RTD (Resistance Temperature Detector) – a temperature sensing element.

RTD gives a linear output and offers a temperature range of 220 degree Celsius to 550 degree Celsius. This RTD is immersed into the water tank of the machine. As the temperature within the tank varies, the resistance of the RTD also varies. This change in the temperature is converted to a corresponding change in resistance.

The RTD is connected across a bridge network as shown in figure. As long as the water temperature is equal to the set temperature, the bridge is balanced and the output of the differential amplifier in the IC LM324 – Low Power Quad Operational Amplifier is zero. If there is a variation in the temperature, the bridge becomes unbalanced immediately and a differential voltage is obtained at the output of the differential amplifier. This differential output is an analog signal and must be digitized before being fed to the microcontroller for the comparison to take place. The ADC0809 is used for this purpose.

3.2 Analog to digital converter

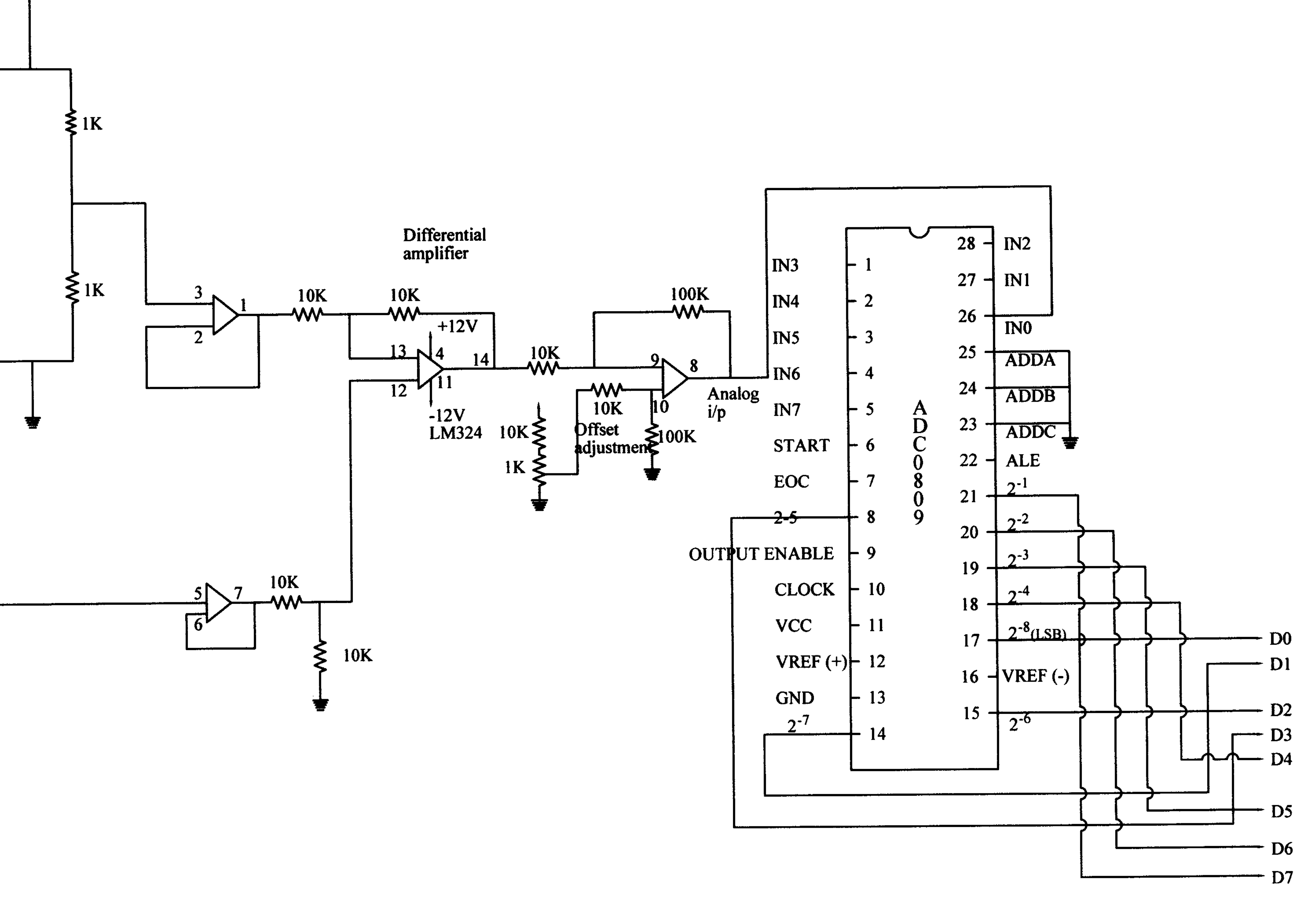
The ADC0809 is an 8-bit microprocessor compatible A/D converter with an 8 channel multiplexer. It uses the successive approximation technique for conversion having a resolution of 8 bits. The A/D output range is from 00 to FF, for an input of 0 to 5 Volts. Any one of the 8 input channels (INO - IN7) can be selected with the help of three address lines (ADD A, ADD B, ADD C). The conversion time is 100 microseconds and hence this converter is highly accurate high speed device well suited for control and machine applications.

Interfacing the A/D Converter to the Microcontroller

Successive Approximation A/D Converters usually have outputs for each bit. The code output to these lines is usually straight binary or offset binary. The 8 parallel outputs of the converter (D0 – D7) can be directly connected to any of the 8 input port pins. there are two other successive approximation A/D converter signal lines that are required to interface the converter to the microcontroller.

1. START CONVERT – this is the signal output from the microcontroller to the A/D converter to tell it to start the conversion process.
2. STATUS – this is the signal which is output by the A/D converter to indicate that the conversion is complete and the word on the outputs is valid.

Thus, the difference in temperature sensed within the tank is fed to one of the 8 inputs (IN4) of the A/D converter. This signal is converted to a corresponding digital value and fed to the microcontroller.



1K
1K
Differential amplifier
3
2
1
10K
10K
+12V
13
4
14
12
11
-12V
LM324
10K
100K
9
8
10K
10
Analog i/p
10K
1K
Offset adjustment
100K
5
7
6
10K
10K
IN3
IN4
IN5
IN6
IN7
START
EOC
2-5
OUTPUT ENABLE
CLOCK
VCC
VREF (+)
GND
2-7
ADC0809
1
2
3
4
5
6
7
8
9
10
11
12
13
14
28
27
26
25
24
23
22
21
20
19
18
17
16
15
IN2
IN1
IN0
ADDA
ADDB
ADDC
ALE
2-1
2-2
2-3
2-4
2-8(LSB)
VREF (-)
2-6
D0
D1
D2
D3
D4
D5
D6
D7

3.3 Microcontroller AT89C51

The AT89C51 microcontroller controls the entire process involved in this project.

Inputs to 89C51 are :

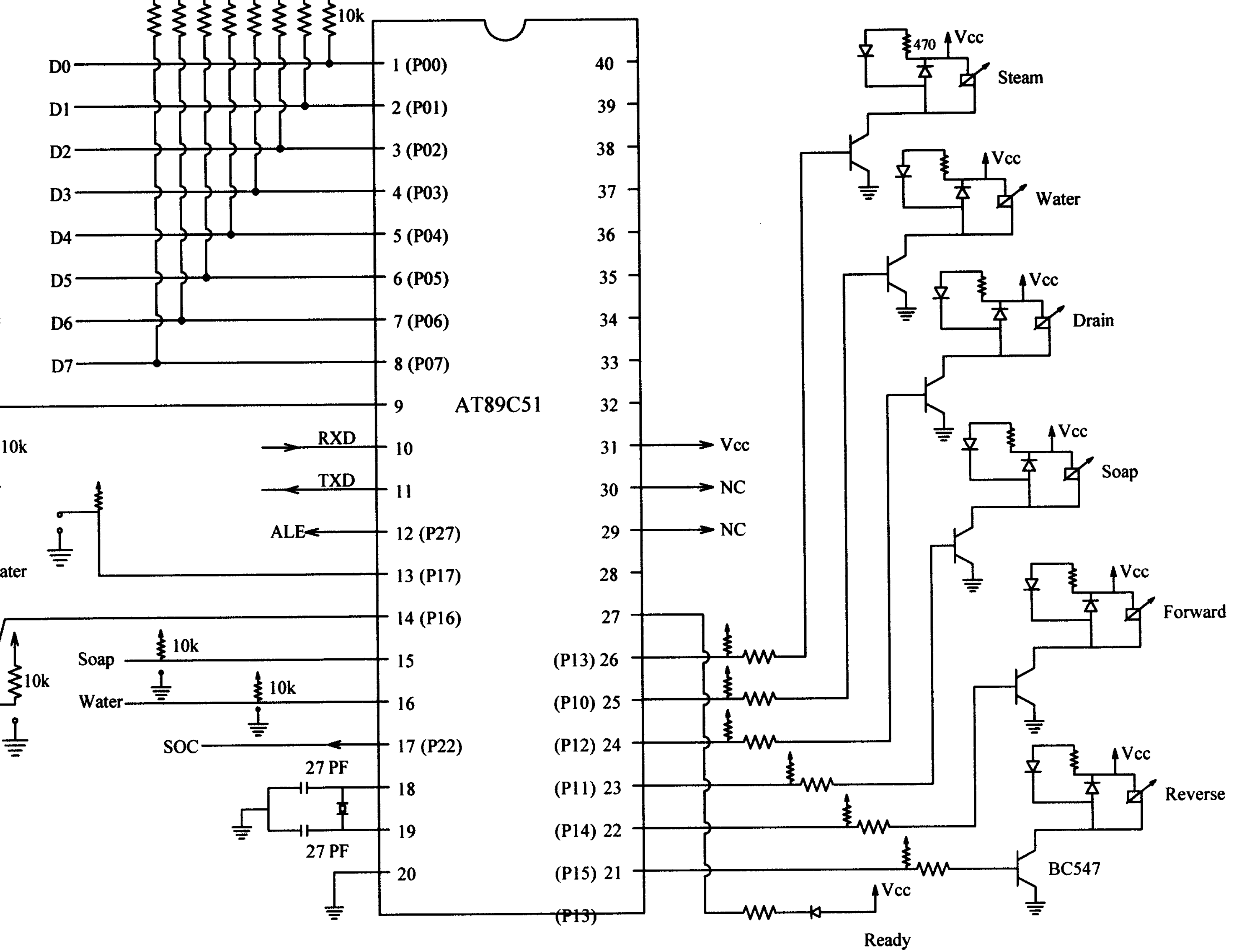
- ADC outputs D0 – D7 (temperature control).
- Water level sensor.
- Manual switch control for soap water solution.
- Start and stop control.

Microcontroller controls the following :

- Forward relay solenoid.
- Reverse relay solenoid.
- Water inlet solenoid.
- Steam inlet solenoid.
- Soap water solenoid.
- Drain outlet solenoid.

Initially when start button of the required machine is pressed in the PC, the communication with that particular machine is enabled by that microcontroller. When 1/3rd of the tank is full, the sensed signal is fed to the microcontroller. The microcontroller then automatically turns of the water solenoid and turns on the soap solenoid. The signal from the manual switch required to turn OFF the soap solenoid is fed to the microcontroller.

In order to maintain a constant set water temperature, the microcontroller continuously compares the water temperature with the set temperature throughout the process. If the set temperature is greater than the water temperature, the microcontroller turns ON the steam solenoid so that the water temperature increases. Once the water temperature reaches the set temperature, the microcontroller automatically turns OFF the steam solenoid. The microcontroller controls the forward and reverse rotation of the motor according to the timings set in the PC. After every forward reverse cycle, the microcontroller checks if the total time is over. If not this cycle continues until the total time becomes zero. Once the total time is over, the process stops and the microcontroller turns ON the drain solenoid. In case the machine is needed to be switched OFF while the process is going on, the stop button on the front panel of the PC is pressed and the process stops abruptly.

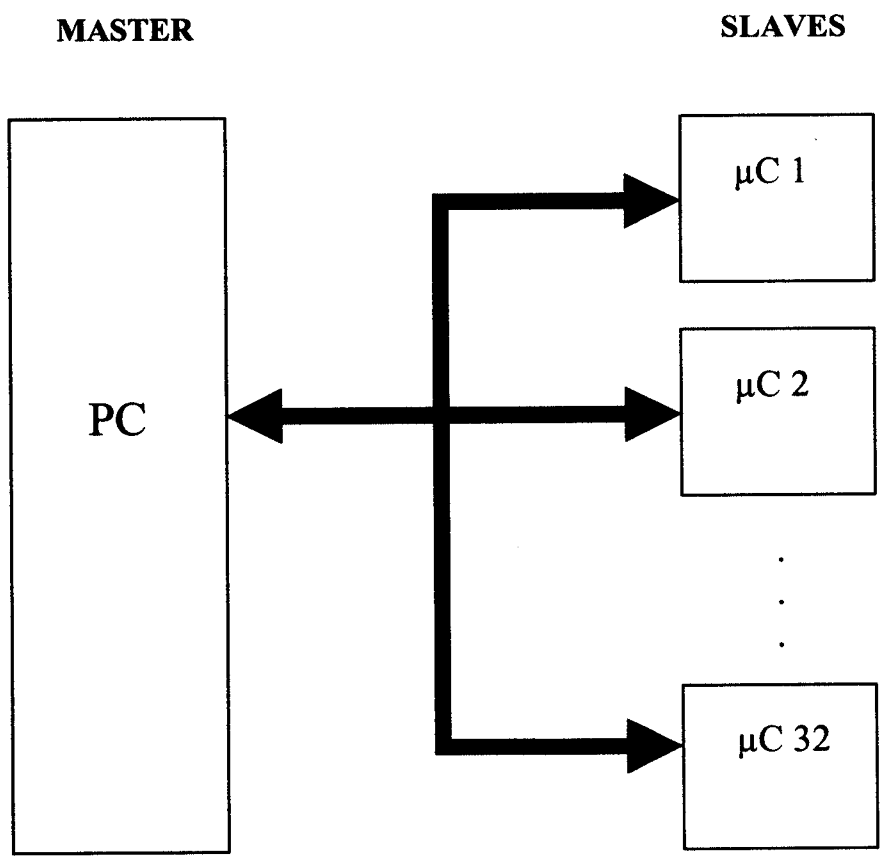


AT89C51
D0
D1
D2
D3
D4
D5
D6
D7
10k
1 (P00)
2 (P01)
3 (P02)
4 (P03)
5 (P04)
6 (P05)
7 (P06)
8 (P07)
9
10
11
12 (P27)
13 (P17)
14 (P16)
15
16
17 (P22)
18
19
20
40
39
38
37
36
35
34
33
32
31
30
29
28
27
(P13) 26
(P10) 25
(P12) 24
(P11) 23
(P14) 22
(P15) 21
(P13)
RXD
TXD
ALE
Soap
Water
SOC
27 PF
27 PF
Vcc
NC
NC
470
Steam
Water
Drain
Soap
Forward
Reverse
BC547
Ready

3.4 RS485 Serial Interface

RS232 standard supports point to point communication. Since several machines are to be interfaced with a PC we go in for the RS485 standard which supports multipoint communication. The RS485 includes the MAX232 and the DS75176 Differential Bus Transceiver.

RS485 Communication

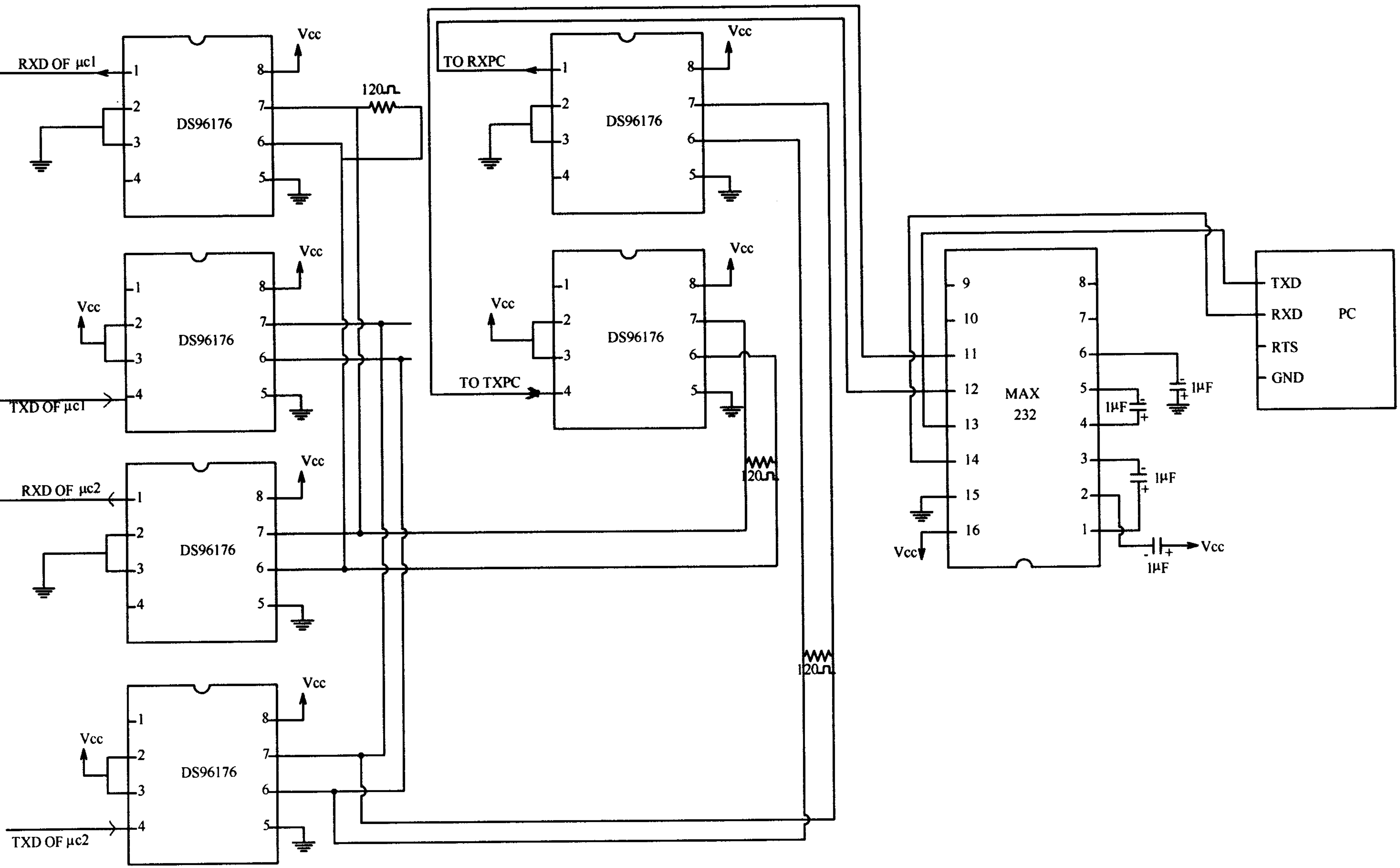


DS75176 – Differential Bus Transceiver consists of a differential line driver and a differential input line receiver both of which have an active enable for the control of the direction. The driver outputs and the receiver inputs are internally connected to form differential I/O bus ports. These port features make the port suitable for multipoint applications in noisy environments.

The driver output of this transceiver is connected to the RXD pin of the microcontroller and the TXD pin of the microcontroller is connected to the receiver input of the transceiver. A control signal from the microcontroller is given to the enable pins of the transceiver. This determines whether the microcontroller is going to transmit or receive.

The RXD and TXD pins of both the microcontrollers are connected to individual transceivers and depending on the address transmitted, the communication between that particular machine and the PC is enabled.

RS 485 COMMUNICATION



COMPONENT DESCRIPTION

4.1 Introduction to Microcontroller

A microcontroller consists of powerful CPU tightly coupled with memory(RAM,ROM or EPROM), various I/O features such as serial ports, parallel ports, timers/counters, interrupt controller, data acquisition interfaces, analog to digital converter, digital to analog converter everything integrated onto a single silicon chip.

Any microcomputer system requires memory to store sequence of instructions making up a program, parallel port or serial port for communicating with an external system, timer/counter for control purposes like generating time delays, baud rate for the serial port, apart from the controlling unit called the central processing unit.

If a system is developed with a microprocessor, the designer has to go for external memory such as RAM, ROM or EPROM and peripherals and hence the size of the PCB will be large. But the microcontroller has got all these peripheral facilities on a single chip. So development of a similar system with a microcontroller reduces PCB size and cost of the design.

4.2 Advantages of microcontroller

If a system is developed with a microprocessor, the designer will have to go for external memories such as RAM, ROM or EPROM and peripherals and hence the size of the PCB will be large enough to hold all the required peripherals. But, the microcontroller has got all these peripheral facilities on a single chip. So development of a similar system with a microcontroller reduces PCB size and cost of the design.

The microcontroller has two 16 bit timers/counters built within it, which makes it more suitable for this application since we need produce some accurate timing delays. It is even more advantages that the timers also act as interrupts.

4.3 Application of Microcontrollers

Microcontrollers are designed for use in sophisticated real time applications such as industrial control, instrumentation and intelligent computer peripherals.

Microcontrollers with ADC find usage in data acquisition system and closed loop analog controllers. It permits considerable system integration by combining analog and digital I/O processing in the single chip.

They are used in industrial applications to control motor, robotics, discrete and continuous process control, in missile guidance and control, in medical instrumentation, oscilloscope, telecommunication, automobiles, for scanning a keyboard, driving an LCD, for frequency measurements, period measurements and so on.

4.4 8-Bit Microcontrollers

Eight-bit microcontrollers represent a transition zone between the dedicated, high-volume, 4-bit microcontrollers and the high-performance, 16 and 32-bit units.

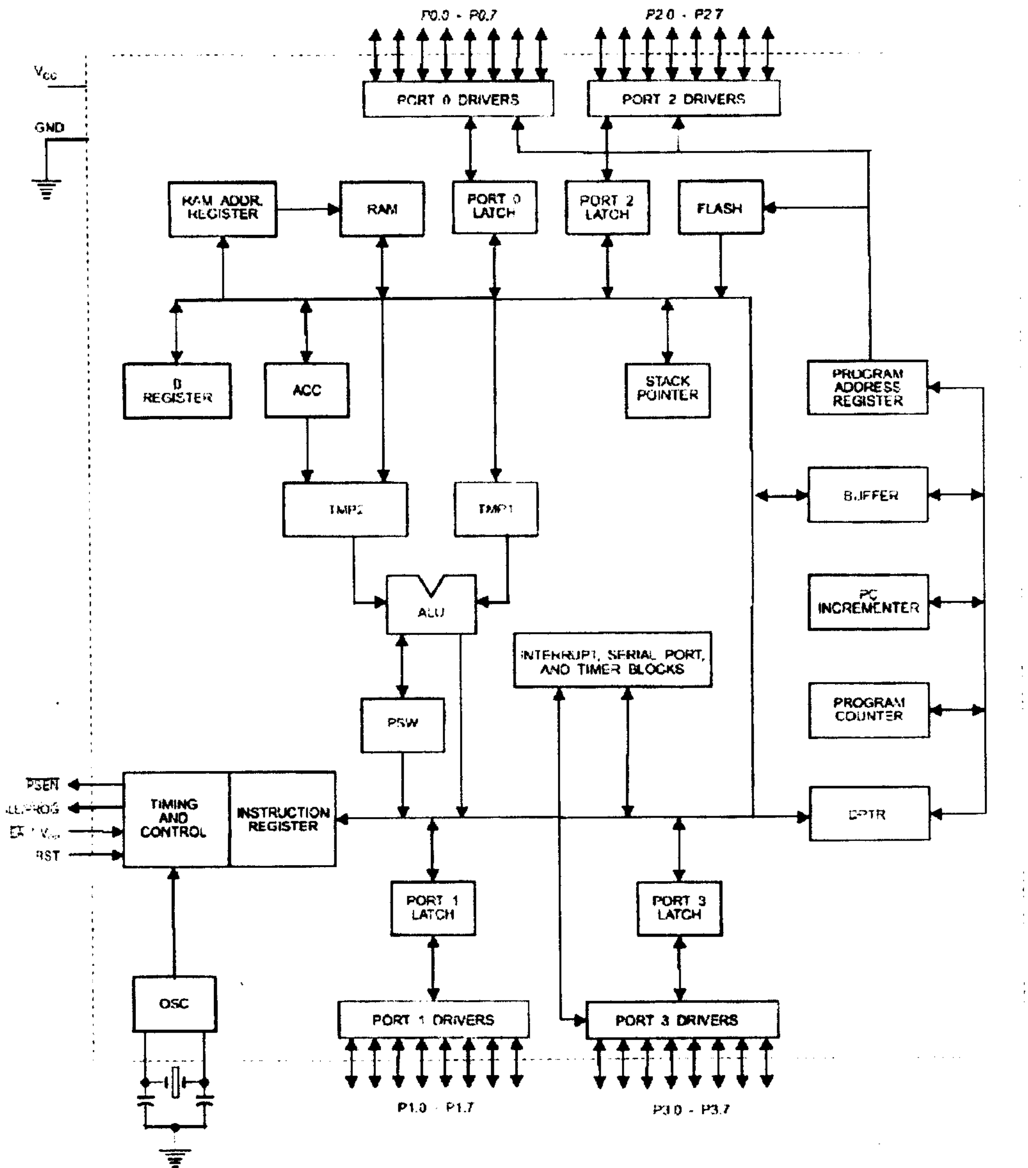
Eight bits has proven to be a very useful word size for small computing tasks. Capable of 256 decimal values or quarter-percent resolution, the 1-byte word is adequate for many control and monitoring applications. Serial ASCII data is also stored in byte sizes, making 8 bits the natural choice for data communications. Most integrated circuit memories and many logic functions are arranged in an 8bit configuration that interfaces easily to data buses of 8 bits.

4.5 AT89C51 8-Bit Microcontroller

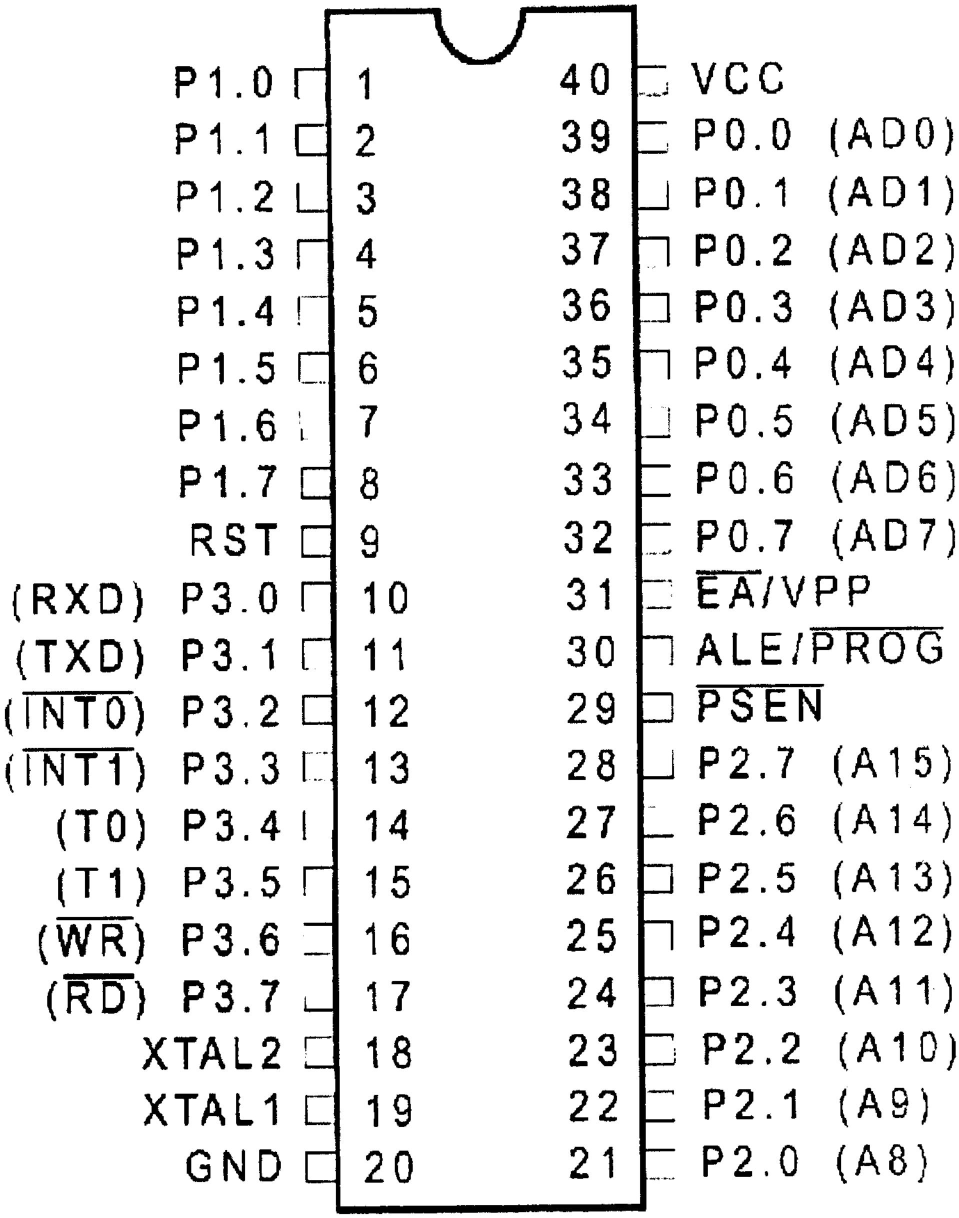
4.5.1 Features

- 8-Bit CPU optimized for control applications
- 4 Kbytes of In-System Reprogrammable Flash Memory
- 128 x 8-Bit Internal RAM
- Bidirectional and Individually Addressable 32 Programmable I/O lines
- Two 16-Bit Timer/Counters
- Extensive Boolean Processing Capabilities (Single-Bit Logic)
- Full Duplex UART
- Multiple Source/Vector/Priority Interrupt Structure
- On-Chip Clock Oscillator
- On-Chip EEPROM
- SPI Serial Bus Interface
- Watchdog Timer

4.5.2 AT89C51 ARCHITECTURE



4.5.3 PIN CONFIGURATION



4.5.4 Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcontroller with 4 Kbytes of Flash Programmable and Erasable Read Only Memory (EPROM). 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.

The AT89C51 provides the following standard features; 4 Kbytes 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. 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. In this application it uses strong internal pull-ups when emitting 1s. during accesses to external data memory that use 8-bit addresses , 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 0:

Port 0 is an 8-bit open drain bi-directional I/O port. As an output port each pin can sink 8 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 pull-ups. Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pull-ups are required during program verification.

Port 1:

Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL input. When 1s are written to Port 1 pins they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current because of the internal pull-ups. Port 1 also receives the low-order address bytes during Flash programming and program verification.

Port 2:

Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. 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 pull-ups and can be used as inputs. As inputs, port 2 pins that are externally being pulled low will source current because of the internal pull-ups. 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. In this application it uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses, 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 pull-ups. 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 pull-ups and can be used as inputs. As inputs, port 3 pins that are externally being pulled low will source current because of the internal pull-ups. 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	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 0 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

4.5.5 Memory organization

The 8051 maintains separate address spaces for program memory and data memory. The program memory can be up to 64 KB, of which the lowest 4 KB are in the on-chip ROM. The data memory consists of 128B of on-chip RAM, plus 21 special function registers, in addition to which the device is capable of accessing up to 64 KB of external data memory.

The program memory uses 16 bit addresses. The external data memory can use either 8 bit or 16 bit addresses. The internal data memory uses 8 bit addresses, which provide a 256 location address space. The lower 128 addresses access the on-chip RAM. The SFRs occupy various locations in the upper 128 bytes of the same address space.

The lowest 32 bytes in the internal RAM are divided into 4 banks of registers, each consisting of 8 bytes. Any of these can be selected to be the "working registers" of the CPU and can be accessed by a 3-bit address in the same byte as the opcode of an instruction.

The next higher 16 bytes of the internal RAM have individually addressable bits. These are provided for use as software flags or for 1 bit(Boolean) processing. This bit addressing capability is an important feature of the 8051.In addition to the 128 individually addressable bit in RAM, 11 of the SFRs also have individually addressable bits.

4.5.6 Serial Data Buffer

The serial buffer is actually two separate registers. When data is moved to SBUF, it goes to the transmit buffer, when it is held for serial transmission. When data is moved from SBUF, it comes from the receive buffer. During serial reception, the incoming bits are clocked into a separate shift register. When reception of a frame is complete and if various other conditions are satisfied, 8 received data bits are transferred from the shift register to the receive buffer. The shift register is then ready to commence reception of a second frame, while the frame already received awaits servicing.

4.5.7 Serial Port Data Registers

In all serial ports a write to SBUF loads the same 9-bit shift register. The data byte goes into the first 8 bits, with the LSB at the output bit of the register. The write to SBUF also loads the 9^{th} bit of the shift register with either a 1 or TB8, depending on the mode and it initiates the transmission.

The receive registers form an input shift register which is 8 bits wide in mode 0 and 9 bits wide in the other modes, SBUF itself, a read-only register which is loaded by the hardware with the data byte at the same time that R1 is activated. In the UART modes, the 9^{th} bit is loaded into RB8 in SCON at the same time the data byte is loaded into SBUF. RB8 and SBUF are not changed if SM2 causes the received data to be ignored.

4.5.8 Serial Interface

The serial port is full duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before previously received byte has been read from the receive register. The serial port registers are both accessed at SFR SBUF. A write to SBUF loads the transmit register and a read accesses a physically separate receive register.

4.5.9 Counters and Timers

Many microcontroller applications require the counting of external events, such as the frequency of the pulse train, or the generation of précis internal time delays between computer actions. Both of these tasks can be accomplished during software techniques, but software loops for counting or timing keep the processor occupied so that, other more important, functions are not done. To relieve the processor from this burden, two 16-bit up counters named T0 and T1, are provided for the general use of the programmer. Each counter may be programmed to count the internal clock pulses, acting as a timer, or programmed to count external pulses as a counter.

In the 'timer' function the register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator period , the count rate is 1/12 of the oscillator frequency. In the 'counter' the register is incremented in response 1-to-0 transition corresponding to its external input pin, T0 or T1. In this function the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes 2 machine cycles to recognize a 1-to-0 transition , the maximum count rate is 1/24 of the oscillator frequency.

4.6 RS 485 COMMUNICATION

4.6.1 RS 232 Standard

RS232 was developed in the 1960s, and among other things, specified an electrical standard, a protocol standard, handshaking, and connector pin-out. In general, many current applications for RS232 use only the electrical standard (3-wires, TDX, RXD, Common) and connector pin-out. While handshaking is still with us, it is usually best to disable it in software (if possible) and/or "loop-back" the pairs of signals (RTS to CTS, DTR to DSR, etc.). While RS232 was rumored to be on the "way out" with the advent of many of the new communications standards, it is still alive and well today. While the standard only supports low data rates and short line length (50ft.) it is still widely used and, very useful in many applications. But only point to point communication is possible using RS232. Hence for a multipoint communication we use the external converters (RS232⇔RS485). With an external converters (RS232⇔RS485) many of the limitations of RS232 can be improved, to take advantage of, the superior properties of differential communications (2-wire or 4-wire).

4.6.2 RS485 Standard

The RS485 standard addresses the problem of data transmission, where a balanced (differential) transmission line is used in a multi-drop (party line) configuration (or point-to-point if only two devices are on the network). Up to 32-nodes (drivers and receivers) are allowed on one multi-drop, bi-directional network. Data rates of up to 10M bps are supported over short distances (40ft.). At the four-thousand foot distance limit, data rates of up to 100K bps are allowable. RS485 specifies a 2-wire, half-duplex communications bus. While RS485 is a 2-wire standard, it does offer 32 nodes on a network, on the other hand RS422 (a 4-wire standard) only specifies up to 10 nodes. Therefore, while not technically correct, it does make some sense to refer to a 4-wire RS485 network that would extend the number of nodes on a 4-wire network to 32 standard loads.

The RS485 standard only specifies electrical characteristics of the driver and the receiver, it does not specify or recommend any protocol. Because matters of protocol are left to the user, it is often difficult (if not impossible) to connect RS485 devices from different manufacturers on the same network.

The RS485 standard allows the user to configure inexpensive local networks and multipoint communications links using twisted pair wire. A typical RS485 network can operate properly in the presence of reasonable ground differential voltages, withstand driver contentious situations, provide reliable communications in electrically noisy environments (good common mode rejection using twisted pair cable, shielding provides additional protection), and support thirty-two or more (many IC manufacturers have 1/2, 1/4, 1/8 unit load devices) drivers and receivers on the line.

4.6.3 DS75176 Differential Bus Transceiver

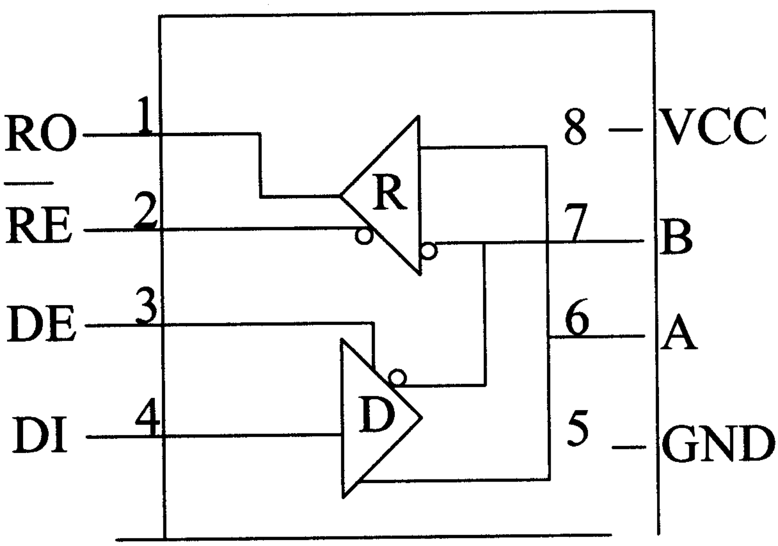
Converters in general can be used to change the electrical characteristic of one communications standard into another, to take advantage of the best properties of the alternate standard selected. For example, a RS232<=>RS485 converter, could be connected to a computer's RS232, full-duplex port, and transform it into an RS485 multipoint network at distances up to 4000ft.

The DS75176 Differential Bus Transceiver is a monolithic integrated circuit designed for bi-directional data communication on balanced multipoint bus transmission lines.

Features

- Bi-directional transceiver.
- Individual driver and receiver enables.
- The transceiver meets EIA standard RS485 and RS422.
- Operates from a single 5V supply.
- Wide positive and negative input and output bus voltage ranges.
- Low power requirement

DS75176 – Connection Diagram



RO –RECEIVER OUTPUT.
RE –RECEIVER ENABLE.
DE –DRIVER ENABLE.
DI –DRIVER INPUT.
A,B –IN/OUT BUS PORT

4.6.4 MAX 232

Features

- Operate from Single +5V Power Supply (+5V and +12V-MAX231/MAX239)
- Low-Power Receive Mode in Shutdown (MAX223/MAX242)
- Meet All EIA/TIA-232E and V.28 Specifications
- Multiple Drivers and Receivers
- 3-State Driver and Receiver Outputs
- Open-Line Detection (MAX243
- ESD Protection for RS232 I/O pins
- Latch up Free (unlike bipolar equivalents)

General Description

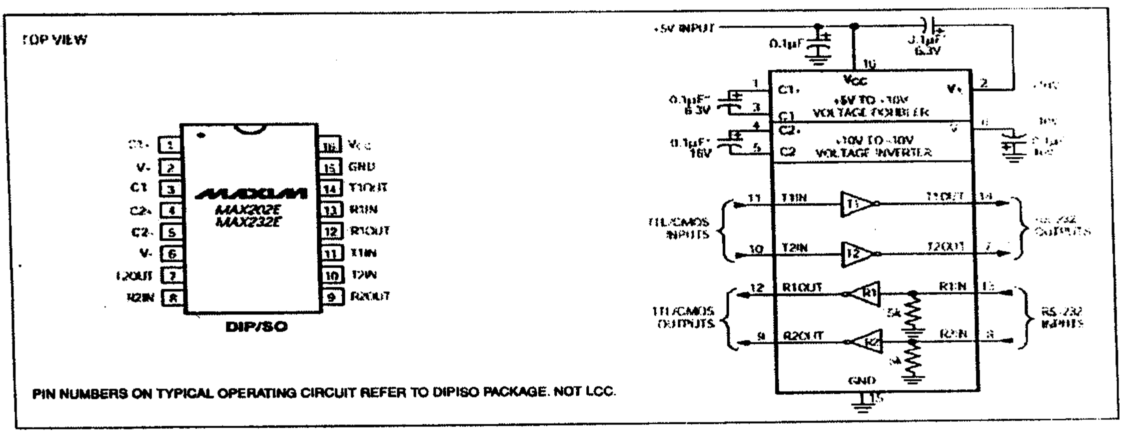
The MAX220-MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E communications interfaces, particularly applications where ±12V is not available. These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5μW.

The MAX 232 line drivers/receivers are designed for RS232 communication in harsh environment. Each transmitter output and receiver input is protected against ± 15KV electrostatic discharge (ESD) shocks, without latch up. The drivers and receivers

Applications

- ♦ Portable Computers
- ♦ Low-Power Modems
- ♦ Interface Translation

Pin Configurations and Typical Operating Circuits



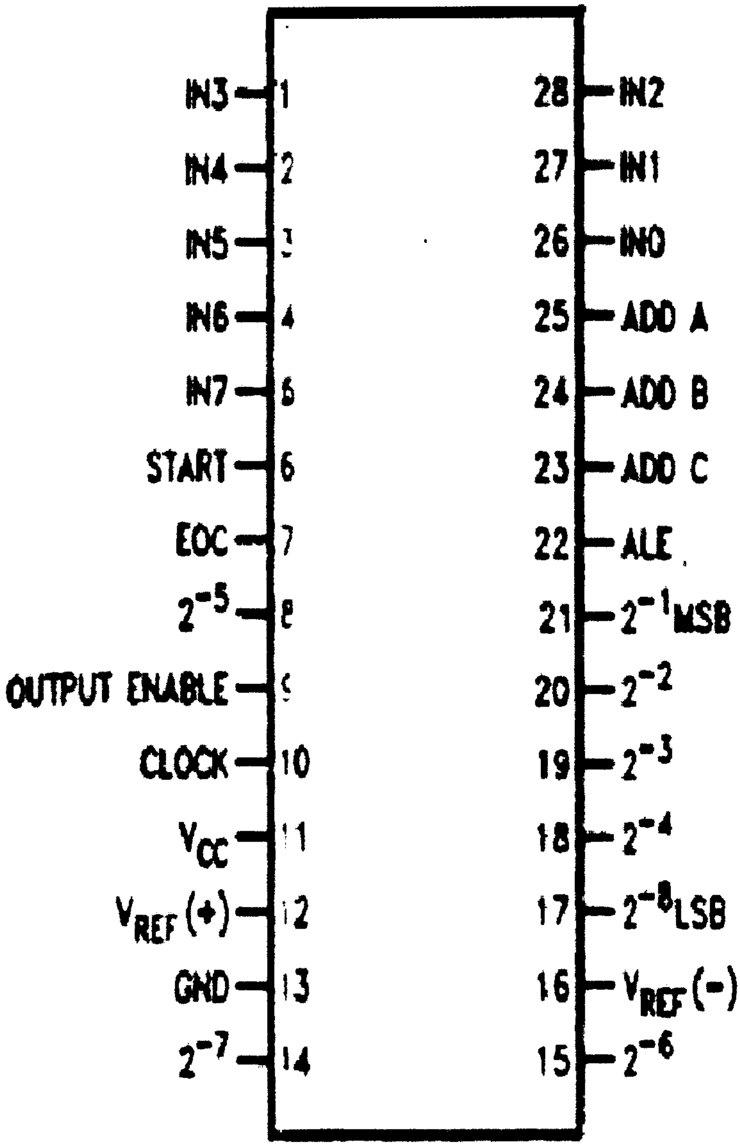
4.7 ADC0809

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.

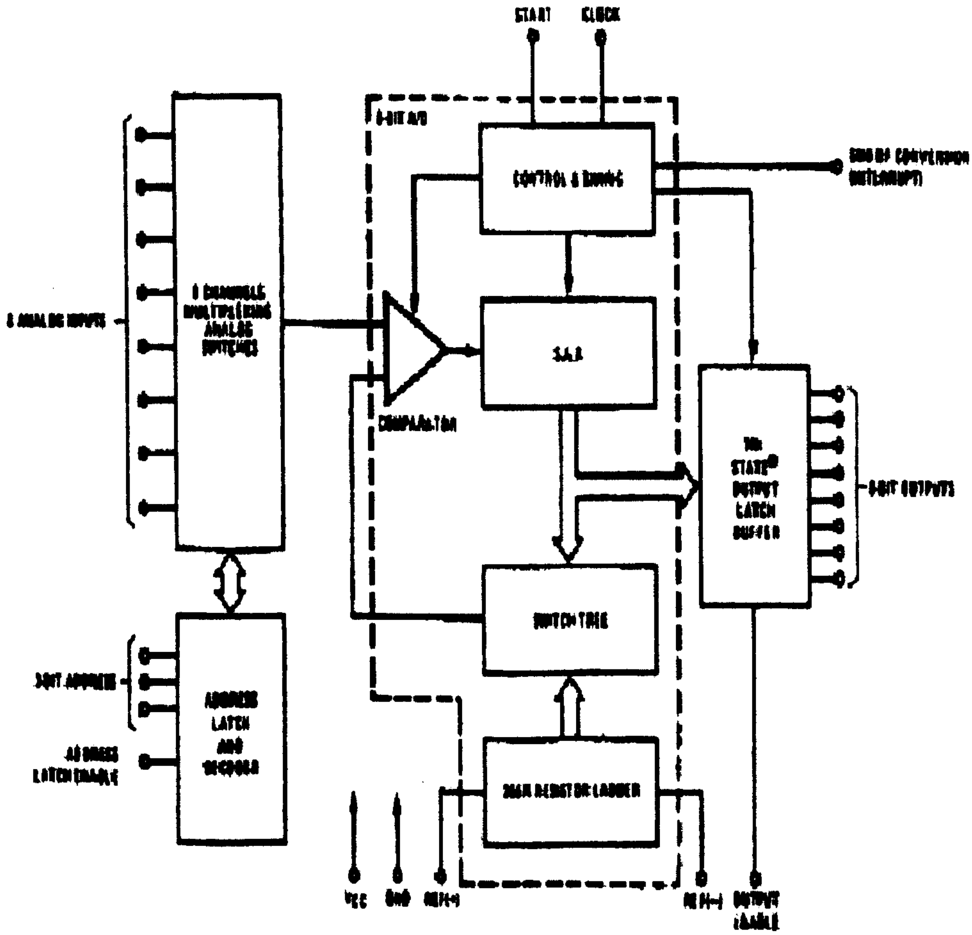
4.7.1 Features

- Easy interface to all microprocessors
- Operates ratiometrically or with 5 V_{DC} or analog span adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to 5V input range with single 5V power supply
- 28-pin molded chip carrier package

4.7.2 ADC0809 CONNECTION DIAGRAM



4.7.3 BLOCK DIAGRAM

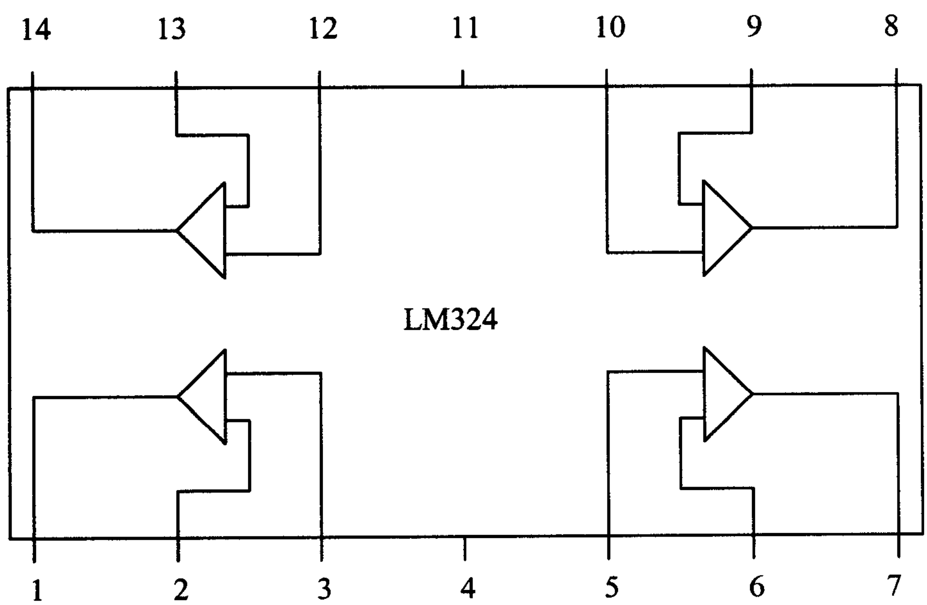


4.8 LM324 LOW POWER QUAD OP AMP

It consists of 4 independent high gain internally frequency compensated op amps which are designed specifically to operate from a single power supply.

4.8.1 FEATURES :

- Eliminates need for dual supplies.
- 4 internally compensated op amps in a single package.
- Compatible with all forms of logic.



SOFTWARE

5 Software description

The software for our project, "MICROCONTROLLER BASED WASHING MACHINE CONTROL THROUGH PC" is written using the C compiler and it is programmed in the 4 K Flash memory. The front panel in the PC and the program for the interface of the PC to the microcontroller is written using Visual Basic 6.0.

5.1 Algorithm

STEP 1:

Start

STEP 2:

Set the timings and water temperature in front panel.

STEP 3:

Select the required machine.

STEP 4:

Press the start button of the selected machine.

STEP 5:

Check for reception in the microcontroller.

STEP 6:

If received send an acknowledgement from microcontroller to PC

STEP 7:

Microcontroller turns the water solenoid ON.

STEP 8:

Microcontroller checks for the water level and turns OFF the water solenoid when 1/3rd of the tank is full.

STEP 9:

Soap solenoid automatically turns ON when the water solenoid turns OFF.

STEP 10:

After the required amount of the soap is let in, soap solenoid is manually switched OFF.

STEP 11:

The forward solenoid turns ON and the motor rotates in the forward direction for the timings set in the front panel of the PC.

STEP 12:

Once the forward solenoid turns OFF the motor remains idle for the set time and then the reverse solenoid turns ON.

STEP 13:

The motor rotates in the reverse direction for the set time after which the reverse solenoid turns OFF and the motor remains idle again.

STEP 14:

Then the total process time is checked and if it is equal to zero the process stops and the drain solenoid turns ON. If not the above cycle repeats again till the total time is equal to zero.

STEP 15:

Throughout the process the water temperature must be maintained at a constant value as set in the front panel of the PC.

STEP 16:

The water temperature is compared with the set temperature continuously. If the water temperature is greater than the set temperature the steam solenoid is turned ON. Otherwise the steam solenoid remains OFF.

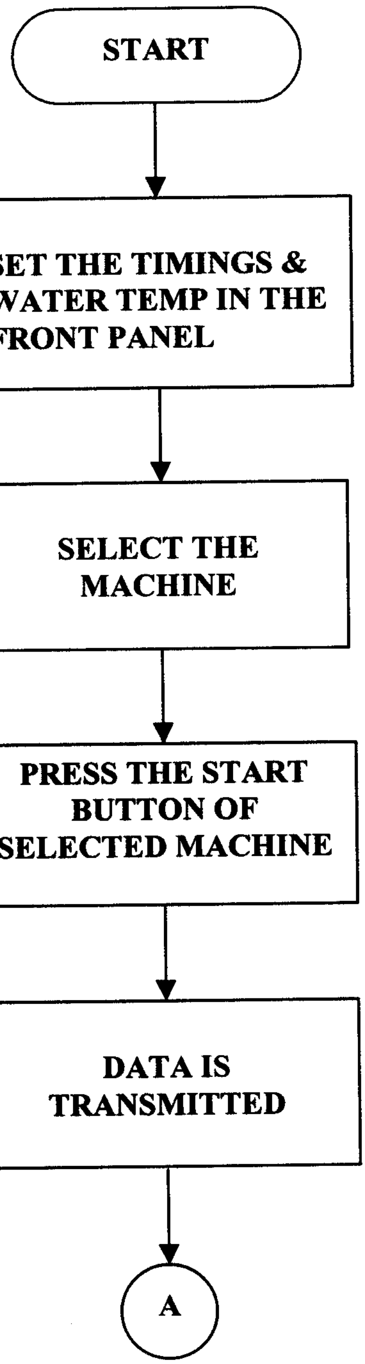
STEP 17:

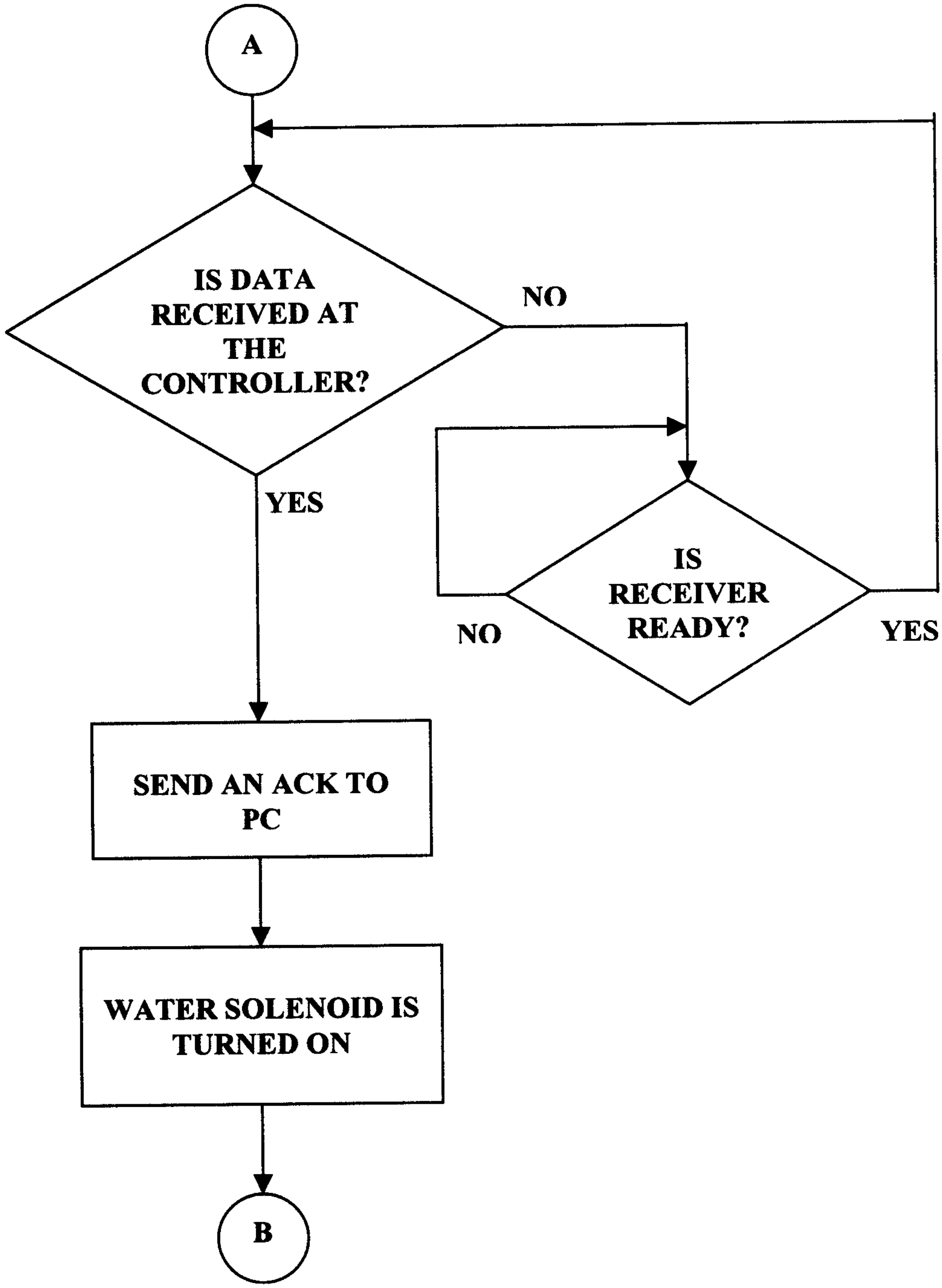
Once the drain solenoid turns ON when the total time is over, the water is let out. Then after a set time, as programmed, the drain solenoid turns OFF and the total process comes to an end.

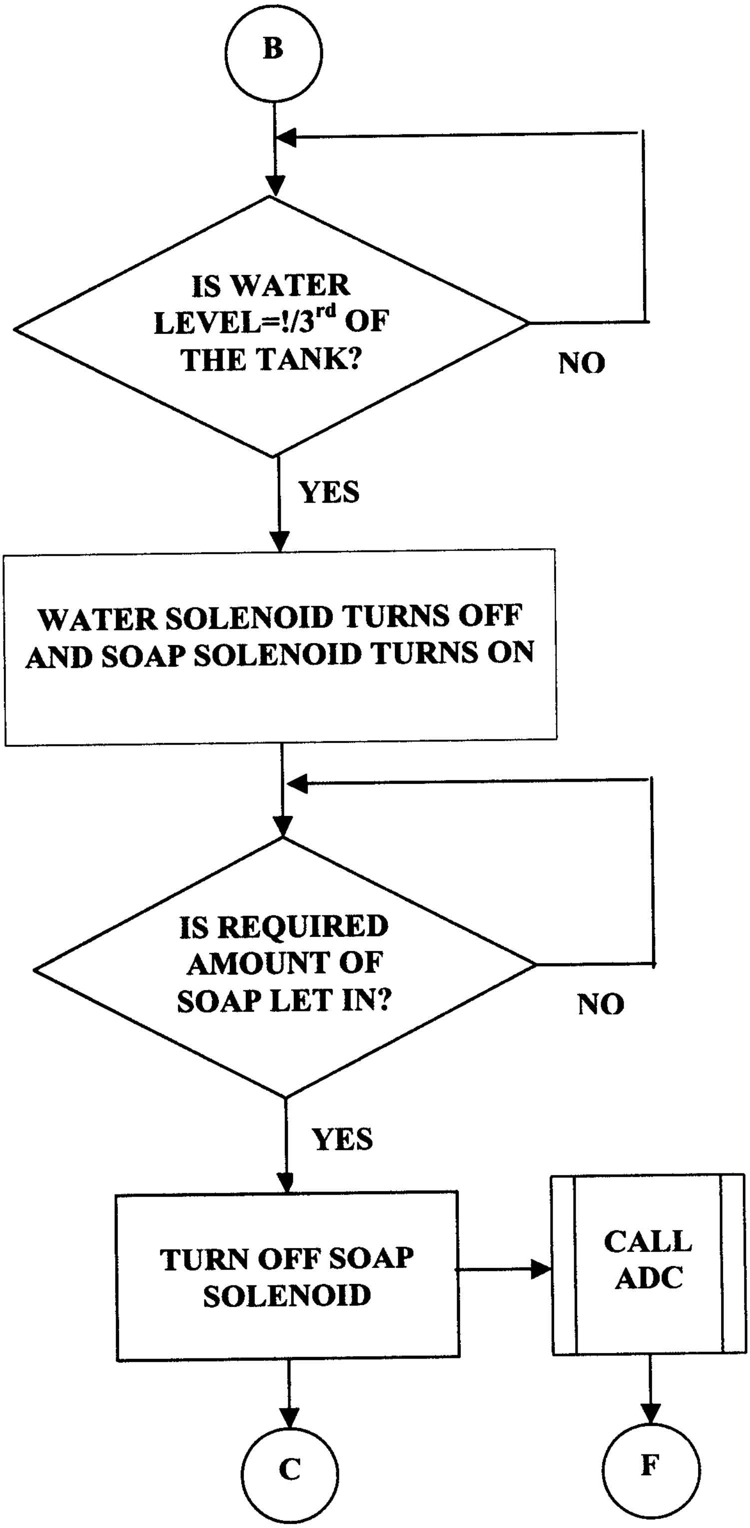
STEP 18:

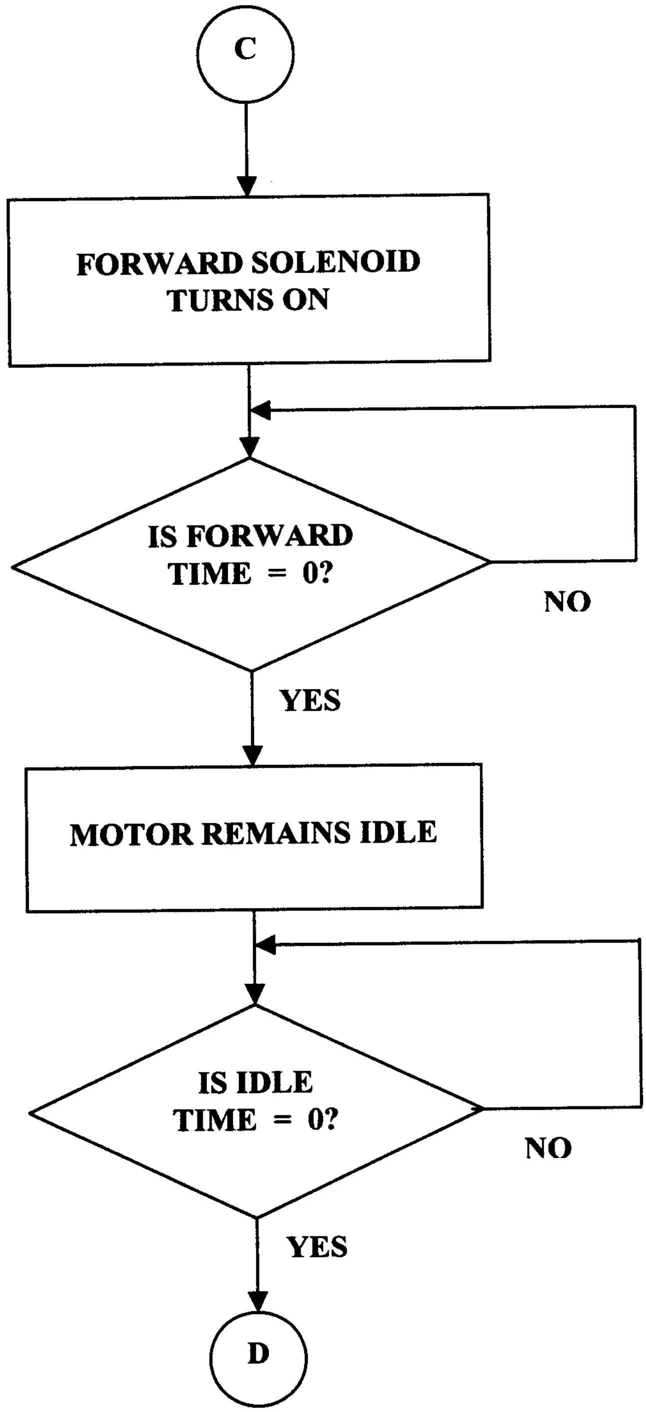
Stop.

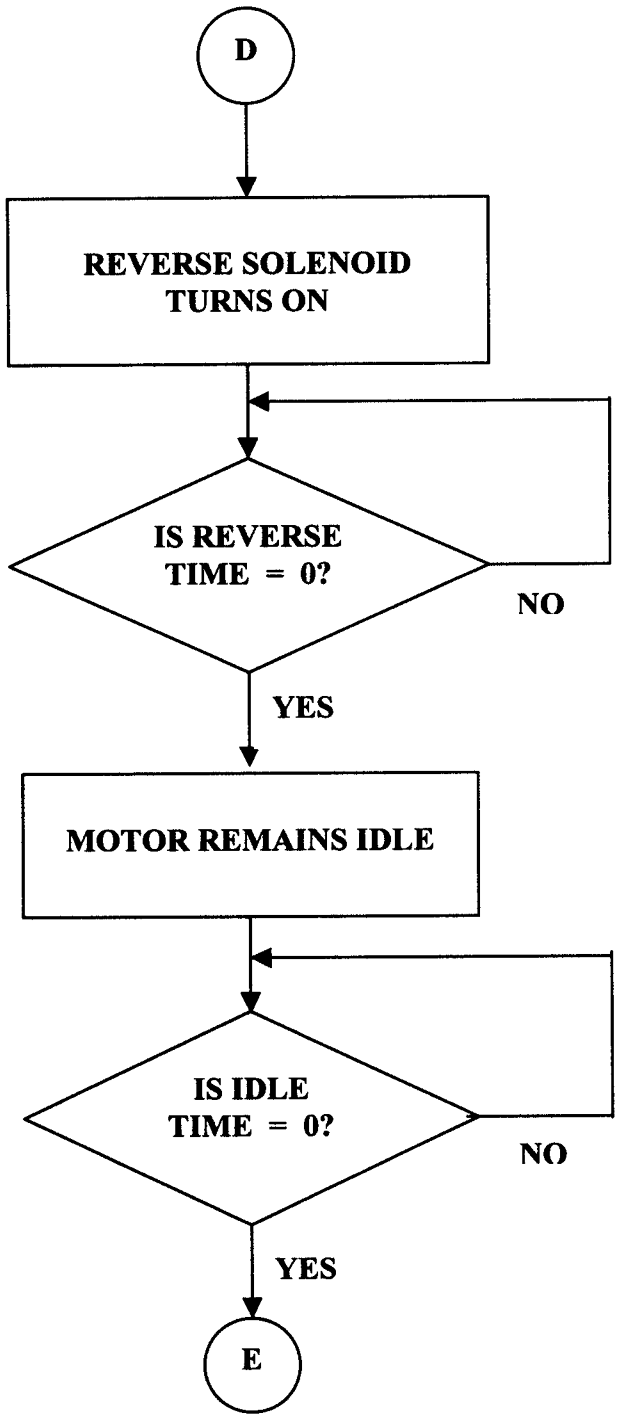
5.2 FLOW CHART

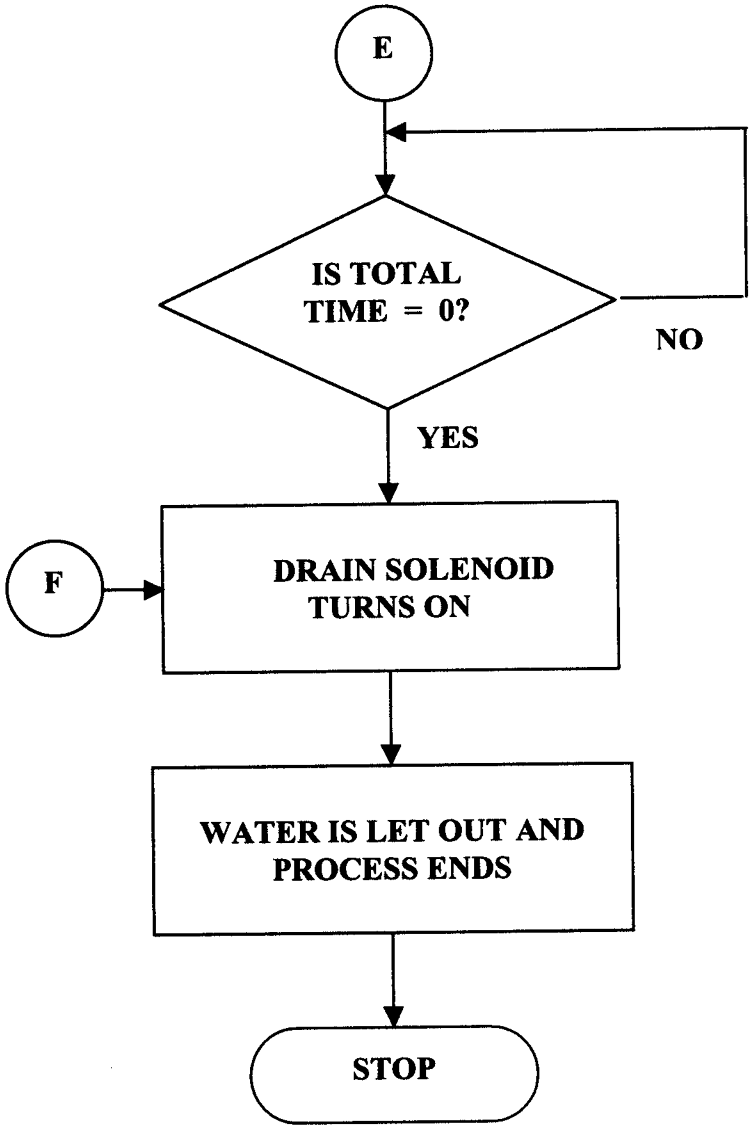



A
IS DATA RECEIVED AT THE CONTROLLER?
NO
YES
IS RECEIVER READY?
NO
YES
SEND AN ACK TO PC
WATER SOLENOID IS TURNED ON
B

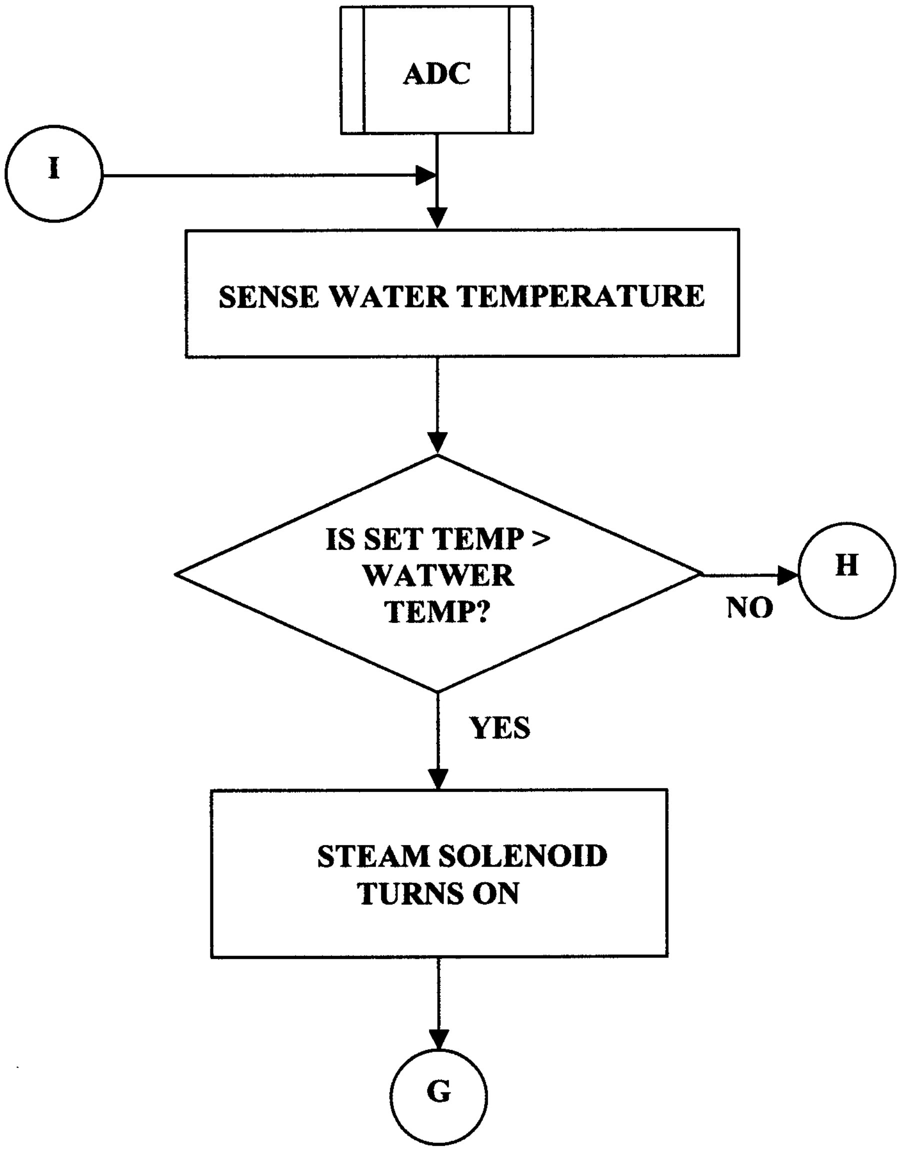

B
IS WATER LEVEL=!/3rd OF THE TANK?
NO
YES
WATER SOLENOID TURNS OFF AND SOAP SOLENOID TURNS ON
IS REQUIRED AMOUNT OF SOAP LET IN?
NO
YES
TURN OFF SOAP SOLENOID
CALL ADC
C
F

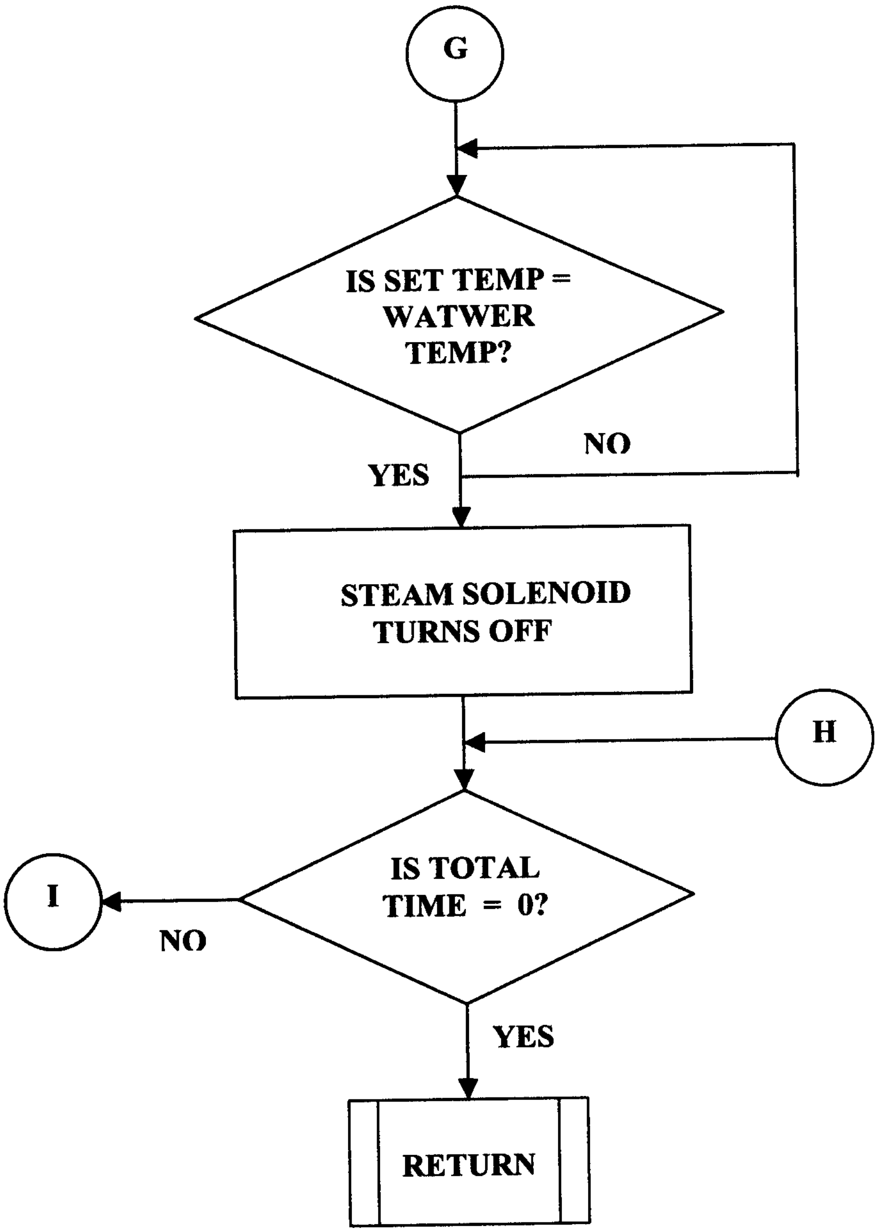

C
FORWARD SOLENOID TURNS ON
IS FORWARD TIME = 0?
NO
YES
MOTOR REMAINS IDLE
IS IDLE TIME = 0?
NO
YES
D


D
REVERSE SOLENOID TURNS ON
IS REVERSE TIME = 0?
NO
YES
MOTOR REMAINS IDLE
IS IDLE TIME = 0?
NO
YES
E


E
IS TOTAL
TIME = 0?
NO
YES
F
DRAIN SOLENOID
TURNS ON
WATER IS LET OUT AND
PROCESS ENDS
STOP

ADC Subroutine




G
IS SET TEMP = WATWER TEMP?
NO
YES
STEAM SOLENOID TURNS OFF
H
IS TOTAL TIME = 0?
I
NO
YES
RETURN

5.3 Program

Visual Basic 6.0 Program

```
Dim outputbuffer As Variant
Dim dataout(0) As Byte
Dim Inputbuffer As Variant
Dim bytearray() As Byte
Dim buffer As Variant
Dim A, B, t As Integer
Dim C, D As Integer
Dim i As Integer

Private Sub Command21_Click()
 If MSComm2.PortOpen = True Then
      MSComm2.PortOpen = False
   End If
   'serial port used COM(1,2,3,4)
   MSComm2.CommPort = 2
   ' 110 baud, no parity, 8 data, and 1 stop bit.
   MSComm2.Settings = "110,N,8,1"
   ' Tell the control to read one byte at a time when Input
   ' is used.
   MSComm2.InputLen = 0
   'transmit and receive buffer size
   MSComm2.OutBufferSize = 256
   MSComm2.InBufferSize = 256
   'Mscomm has no handshaking
   MSComm2.Handshaking = comNone
   MSComm2.EOFEnable = False
   'No oncomm event on received data
   MSComm2.RThreshold = 1
   'No oncomm event on transmit
   MSComm2.SThreshold = 0

   ' Set InputMode to read binary data
   MSComm2.InputMode = comInputModeBinary
   ' Open the port.
   MSComm2.PortOpen = True
   ' Send the attention command to the modem.
  MSComm2.ParityReplace = " "

  transmit1
 End Sub
```

```
Private Sub Command26_Click()
End
End Sub

Private Sub Command3_Click()
If MSComm1.PortOpen = True Then
      MSComm1.PortOpen = False
   End If
   'serial port used COM(1,2,3,4)
   MSComm1.CommPort = 1
   ' 110 baud, no parity, 8 data, and 1 stop bit.
   MSComm1.Settings = "110,N,8,1"
   ' Tell the control to read one byte at a time when Input
   ' is used.
   MSComm1.InputLen = 0
   'transmit and receive buffer size
   MSComm1.OutBufferSize = 256
   MSComm1.InBufferSize = 256
   'Mscomm has no handshaking
   MSComm1.Handshaking = comNone
   MSComm1.EOFEnable = False
   'No oncomm event on received data
   MSComm1.RThreshold = 1
   'No oncomm event on transmit
   MSComm1.SThreshold = 0

   ' Set InputMode to read binary data
   MSComm1.InputMode = comInputModeBinary
   ' Open the port.
   MSComm1.PortOpen = True
   ' Send the attention command to the modem.
  MSComm1.ParityReplace = " "

transmit
End Sub

Private Sub Form_Load()
Dim InString As String
   ' Use COM1.

 End Sub
Private Sub transmit()

dataout(0) = &H81
outputbuffer = dataout()
```

```
MSComm1.Output = outputbuffer
delay
delay

dataout(0) = &HD1
outputbuffer = dataout()
MSComm1.Output = outputbuffer

End Sub

Private Sub transmit1()

dataout(0) = &H81
outputbuffer = dataout()
MSComm2.Output = outputbuffer

delay
delay

dataout(0) = &HD1
outputbuffer = dataout()
MSComm2.Output = outputbuffer

End Sub

Public Sub MSComm1_OnComm()
Dim A As Byte

Dim buffer As Variant
    'Dim Buffer As Variant
    Dim bytesin As Integer
    Dim bytecount As Integer
    Dim iparray() As Byte

Select Case MSComm1.CommEvent
  Case comEvReceive
     i = i + 1

    Do
    DoEvents
    bytesin = MSComm1.InBufferCount
    Loop Until (bytesin >= bytecount)
    buffer = MSComm1.Input
    iparray() = buffer
    A = iparray(0)
```

```
If A = &H99 Then             'water solenoid
    Shape1.FillColor = QBColor(2) 'temperature
    Shape2.FillColor = QBColor(4)
    Shape3.FillColor = QBColor(2)
    Shape7.FillColor = QBColor(4)
    Shape8.FillColor = QBColor(4)
    dataout(0) = Val(Text4.Text)
    outputbuffer = dataout()
    MSComm1.Output = outputbuffer
ElseIf A = &H64 Then
                                        'forward
    dataout(0) = Val(Text5.Text)
    outputbuffer = dataout()
    MSComm1.Output = outputbuffer
ElseIf A = &H65 Then
                                         'reverse
    dataout(0) = Val(Text6.Text)
    outputbuffer = dataout()
    MSComm1.Output = outputbuffer
ElseIf A = &H66 Then
                                         'idle
    dataout(0) = Val(Text7.Text)
    outputbuffer = dataout()
    MSComm1.Output = outputbuffer
ElseIf A = &H67 Then
                                           'TOTAL
    dataout(0) = Val(Text1.Text)
    outputbuffer = dataout()
    MSComm1.Output = outputbuffer
ElseIf A = &H98 Then      'SOAP SOLENOID
    Shape1.FillColor = QBColor(4)
    Shape2.FillColor = QBColor(2)
    Shape3.FillColor = QBColor(2)
    Shape7.FillColor = QBColor(4)
    Shape8.FillColor = QBColor(4)

 ElseIf A = &H97 Then     'SOAP SOLENOID off
    Shape1.FillColor = QBColor(4)
    Shape2.FillColor = QBColor(4)
    Shape3.FillColor = QBColor(2)
    Shape7.FillColor = QBColor(4)
    Shape8.FillColor = QBColor(4)

 ElseIf A = &H96 Then      'PROCESS off
```

```
        Shape1.FillColor = QBColor(4)
        Shape2.FillColor = QBColor(4)
        Shape3.FillColor = QBColor(4)
        Shape7.FillColor = QBColor(4)
        Shape8.FillColor = QBColor(4)

     ElseIf A = &H95 Then      'STEAM SOLENOID ON
        Shape1.FillColor = QBColor(4)
        Shape2.FillColor = QBColor(4)
        Shape3.FillColor = QBColor(2)
        Shape7.FillColor = QBColor(2)
        Shape8.FillColor = QBColor(4)
     ElseIf A = &H94 Then      'STEAM SOLENOID OFF
        Shape1.FillColor = QBColor(4)
        Shape2.FillColor = QBColor(4)
        Shape3.FillColor = QBColor(2)
        Shape7.FillColor = QBColor(4)
        Shape8.FillColor = QBColor(4)
     ElseIf A = &H93 Then      'DRAIM ON
        Shape1.FillColor = QBColor(4)
        Shape2.FillColor = QBColor(4)
        Shape3.FillColor = QBColor(2)
        Shape7.FillColor = QBColor(4)
        Shape8.FillColor = QBColor(2)

    End If

    delay
    delay
  End Select
End Sub
Private Sub delay()
For t = 1 To 1000 Step 1

Next t

End Sub

Private Sub MSComm2_OnComm()
Dim A As Byte
Dim buffer As Variant
   'Dim Buffer As Variant
    Dim bytesin As Integer
    Dim bytecount As Integer
    Dim iparray() As Byte
```

```
Select Case MSComm2.CommEvent
  Case comEvReceive
     i = i + 1

    Do
    DoEvents
    bytesin = MSComm1.InBufferCount
    Loop Until (bytesin >= bytecount)
    buffer = MSComm2.Input
    iparray() = buffer
    A = iparray(0)

    If A = &H99 Then           'water solenoid
        Shape4.FillColor = QBColor(2) 'temperature
        Shape5.FillColor = QBColor(4)
        Shape6.FillColor = QBColor(2)
        Shape9.FillColor = QBColor(4)
        Shape10.FillColor = QBColor(4)
        dataout(0) = Val(Text2.Text)
        outputbuffer = dataout()
        MSComm2.Output = outputbuffer
    ElseIf A = &H64 Then
                                      'forward
        dataout(0) = Val(Text3.Text)
        outputbuffer = dataout()
        MSComm2.Output = outputbuffer
    ElseIf A = &H65 Then
                                       'reverse
        dataout(0) = Val(Text8.Text)
        outputbuffer = dataout()
        MSComm2.Output = outputbuffer
    ElseIf A = &H66 Then
                                       'idle
        dataout(0) = Val(Text9.Text)
        outputbuffer = dataout()
        MSComm2.Output = outputbuffer
    ElseIf A = &H67 Then
                                         'TOTAL
        dataout(0) = Val(Text10.Text)
        outputbuffer = dataout()
        MSComm2.Output = outputbuffer
    ElseIf A = &H98 Then       'WATER SOLENOID ON
        Shape4.FillColor = QBColor(4)
        Shape5.FillColor = QBColor(2)
```

```
        Shape6.FillColor = QBColor(2)
        Shape9.FillColor = QBColor(4)
        Shape10.FillColor = QBColor(4)
     ElseIf A = &H97 Then     'SOAP SOLENOID off
        Shape4.FillColor = QBColor(4)
        Shape5.FillColor = QBColor(4)
        Shape6.FillColor = QBColor(2)
        Shape9.FillColor = QBColor(4)
        Shape10.FillColor = QBColor(4)

     ElseIf A = &H96 Then      'PROCESS off
        Shape4.FillColor = QBColor(4)  'forward
        Shape5.FillColor = QBColor(4)
        Shape6.FillColor = QBColor(4)
        Shape9.FillColor = QBColor(4)
        Shape10.FillColor = QBColor(4)

     ElseIf A = &H95 Then      'STEAM SOLENOID ON
        Shape4.FillColor = QBColor(4)
        Shape5.FillColor = QBColor(4)
        Shape6.FillColor = QBColor(2)
        Shape9.FillColor = QBColor(2)
        Shape10.FillColor = QBColor(4)
     ElseIf A = &H94 Then      'STEAM SOLENOID OFF
        Shape4.FillColor = QBColor(4)
        Shape5.FillColor = QBColor(4)
        Shape6.FillColor = QBColor(2)
        Shape9.FillColor = QBColor(4)
        Shape10.FillColor = QBColor(4)
     ElseIf A = &H93 Then      'DRAIM ON
        Shape4.FillColor = QBColor(4)
        Shape5.FillColor = QBColor(4)
        Shape6.FillColor = QBColor(2)
        Shape9.FillColor = QBColor(4)
        Shape10.FillColor = QBColor(2)
     End If

     delay
     delay
   End Select
End Sub
```

C - Compiler Coding

```
/*  serial data transfer program  at 110 bit baud rate */
/* data is transmitted by pressing a key*/
/*received data is displayed thru port 0*/
#include "stdio.h"
#include "reg51.h"
# define TV0  0xfe //TIMER O INTERRUPT CALL
# define TVL0 0x64
#define ADC_PORT P0

bit process_flag;
bit f_flg;
bit r_flg;
bit ir_flg;
bit if_flg;
bit t_flg;

sbit ALE =P27;
sbit SOC=P22;

bit flagt;
bit flagr;
bit rec;
void delay();
void adc();
unsigned char
rec_reg,forward_time,set_temp,reverse_time,idle_time,total_
time,temp;
unsigned char
dforward_time,dreverse_time,didle_time,dtotal_time,sec,min;
unsigned char d_rec_reg,i,j,temperature,steam_status;
unsigned int misec,t,h;

sbit  water_solenoid = P10;
sbit  soap_solenoid  = P11;
sbit  drain_solenoid = P12;
sbit  steam_solenoid = P13;
sbit  forward        = P14;
sbit  reverse        = P15;
sbit  water_on       = P16;
sbit  soap_on        = P17;
```

```
sbit txled              = P36;
sbit rxled              = P37;



void main()
{
 PCON=0x0;      //intialise smod in pcon register to low
 IE=0x18;       //interrupt enable for serial and timer1
 IP=0x10;       //interrupt priority for serial
 TMOD=0x20;     //timer1 in auto reload mode
 SCON=0x50;     //intialise scon as 8 bit UART mode with
                   reception being enable
 TL1=0x72;
 TH1=0x72;      // timer1 value for serial interrupt @ 6mhz
 TCON=0x40;     //enabling timer1
 TI=0;          //clearing ti bit in scon register
 RI=0;          //clearing ri bit in scon register
 flagt=0;       //bit addressable flags
 flagr=0;
 rec_reg=0;     //ram register
 // led=0;      //led to indicate transmission is over
 EA=1;          //  enables all interrupts


 Forward          = 0;
 Reverse          = 0;
 water_solenoid = 0;
 steam_solenoid = 0;
 soap_solenoid  = 0;
 drain_solenoid = 0;
 process_flag   = 0;
 water_on       = 1;        //key 0 flows
 soap_on        = 1;        // key 0 flows
 process_flag   = 0;
 temperature    = 0;
 txled          = 1;
 rxled          = 1;
 while(1)
 {
 do{;}while(flagr==0);
 flagr=0;
 if(rec_reg==0x81)
 {
   do{;}while(flagr==0);
```

```
  flagr=0;
  if(rec_reg==0xd1)
  {
      water_solenoid=1;      //water solenoid on

      SBUF=0x99;             //if key is pressed a data is
                             //sent to sbuf
      do{;}while(flagt==0);
      flagt=0;
      txled=0;
  }
}

do{;}while(flagr==0);
flagr=0;
set_temp=rec_reg;
rxled=0;


SBUF=0x64;                   //if key is pressed  a data is
                             //sent to sbuf
do{;}while(flagt==0);
flagt=0;
txled=1;

do{;}while(flagr==0);
flagr=0;
forward_time=rec_reg;
rxled=1;


SBUF=0x65;                   //if key is pressed  a data is sent
                             //to sbuf
do{;}while(flagt==0);
flagt=0;
txled=0;

do{;}while(flagr==0);
flagr=0;
reverse_time=rec_reg;
rxled=0;


SBUF=0x66;                   //if key is pressed  a data is sent
                             //to sbuf
```

```
do{;}while(flagt==0);
flagt=0;
txled=1;

do{;}while(flagr==0);
flagr=0;
idle_time=rec_reg;
rxled=1;


SBUF=0x67;                   //if key is pressed  a data is sent
                             //to sbuf
do{;}while(flagt==0);
flagt=0;
txled=0;

do{;}while(flagr==0);
flagr=0;
total_time=rec_reg;
rxled=0;

do{;}while(water_on==1); //tank fill water_on=1
water_solenoid=0;
soap_solenoid=1;

SBUF=0x98;                   //if key is pressed  a data is
                             //sent to sbuf
do{;}while(flagt==0);
flagt=0;

do{;}while(soap_on==1);  //soap poured
soap_solenoid=0;


SBUF=0x97;                   //if key is pressed  a data is
                             //sent to sbuf
do{;}while(flagt==0);
flagt=0;

ALE=1;                       //A high to low pulse of ALE
for(i=1;i<=25;)
{i++;}

SOC=1;                       //A high to low pulse of SOC
for(i=1;i<=25;)
```

```
{i++;}

ALE=0;
for(i=1;i<=25;)

{i++;}
SOC=0;

for(i=1;i<=250;)
{i++;}
temperature=ADC_PORT;


if(set_temp>temperature)
{
   steam_solenoid=1;
   SBUF=0x95;
   do{;}while(flagt==0);
   flagt=0;
}
else
{
     steam_solenoid=0;
     SBUF=0x94;
     do{;}while(flagt==0);
     flagt=0;
}

dforward_time = forward_time;
dreverse_time = reverse_time;
didle_time    = idle_time;
dtotal_time   = total_time;
f_flg     = 0;
r_flg     = 0;
if_flg    = 0;
ir_flg    = 0;
t_flg     = 0;
forward   = 1;
f_flg     = 1;
TR0       = 1;
ET0       = 1;
TMOD      = 0x21;
TL0       = TVL0;
TH0       = TV0;
```

```
do
{
   ALE=1;                    //A high to low pulse of ALE
   for(i=1;i<=25;)
   {i++;}

    SOC=1;                   //A high to low pulse of SOC
    for(i=1;i<=25;)
    {i++;}
    ALE=0;
    for(i=1;i<=25;)

    {i++;}
    SOC=0;

    for(i=1;i<=250;)
    {i++;}
    temperature=ADC_PORT;


    if(set_temp>temperature)
    {
       steam_solenoid=1;
       SBUF=0x95;
       do{;}while(flagt==0);
       flagt=0;
    }
    else
    {
       steam_solenoid=0;
       SBUF=0x94;
       do{;}while(flagt==0);
       flagt=0;
    }

 }while(t_flg==0);

 txled  = 1;
 rxled  = 1;
 TF0    = 0;
 TR0    = 0;
 f_flg  = 0;
 r_flg  = 0;
 if_flg = 0;
 ir_flg = 0;
```

```
t_flg  = 0;
drain_solenoid=1;

SBUF=0x93;
do{;}while(flagt==0);
   flagt=0;


   for (h=1;h<=3000;h++)
   {
      for(t=1;t<=30000;)
      {t++;}

      for(t=1;t<=30000;)
      {t++;}

      for(t=1;t<=30000;)
      {t++;}

      for(t=1;t<=30000;)
      {t++;}

      for(t=1;t<=30000;)
      {t++;}

      for(t=1;t<=30000;)
      {t++;}

      for(t=1;t<=30000;)
      {t++;}
   }

    forward        = 0;
    reverse        = 0;
    water_solenoid = 0;
    steam_solenoid = 0;
    soap_solenoid  = 0;
    drain_solenoid = 0;

    SBUF=0x96;                      //process end
    do{;}while(flagt==0);
    flagt=0;
   }
   }
```

```
void delay()
{
   for(t=1;t<=1000;)
   {t++;}
}


serint ()interrupt 4    //serial vector address
{
   EA=0;
   switch (TI)                 //checks if transmission is over
   {
      case 1:
      {
         TI=0;                 //clears ti flag
         flagt=1;              //sets flagt
         break;
      }
      case 0:
      {
         switch (RI)           //checks if reception is over
         {
           case 0:
           break;
           case 1:
           {
             rec_reg=SBUF;   //received data is fed to
                             //rec_reg from sbuf
             RI=0;           //clears ri flag
             flagr=1;        //sets flagr
             break;}
           }
         }
      }

  EA=1;
}

void timer0() interrupt 1     // Interrupt for real time
clock
    {
       TR1=0;
       TF1=0;
       TR0=0;
```

```
                              // Interrupt looped in 1ms
TH0=TV0;        //0xfe;// Timer 0 16-bit timer
TL0=TVL0;       //0x0b;
misec=misec+1;
if(misec==1800)
   {

        sec=sec+1;
        misec=0;
        txled=~txled;
        if(sec==60)
        {
          min=min+1;
          sec=0;
          misec=0;
          if(total_time==0)
          {
             t_flg   = 1;
             f_flg   = 0;
             r_flg   = 0;
             if_flg  = 0;
             ir_flg  = 0;
             min     = 0;
             misec   = 0;
             sec     = 0;
             forward= 0;
             reverse= 0;
             total_time=dtotal_time;
             t_flg   = 1;
          }
             total_time=total_time-1;
      }

        if(f_flg==1)
        {
          if(forward_time==0)
          {
            f_flg   = 0;
            if_flg  = 1;
            r_flg   = 0;
            ir_flg  = 0;
            misec   = 0;
            forward_time=dforward_time;
          }
          if(f_flg==1)
```

```
  {
    forward_time=forward_time-1;
    rxled    =~rxled;
    forward = 1;
    reverse = 0;
   }
}

else if(if_flg==1)
{
   if(idle_time==0)
   {
       f_flg   = 0;
       if_flg  = 0;
       r_flg   = 1;
       ir_flg  = 0;
       misec   = 0;
       idle_time=didle_time;
   }
   if(if_flg==1)
   {
      idle_time=idle_time-1;
      forward=0;
      reverse=0;
   }
}

else if(r_flg==1)
{
   if(reverse_time==0)
   {
      f_flg=0;
      if_flg=0;
      r_flg=0;
      ir_flg=1;
      misec=0;
      reverse_time=dreverse_time;
   }
   if(r_flg==1)
   {
      reverse_time=reverse_time-1;
      forward=0;
      reverse=1;
   }
}
```

```
                else if(ir_flg==1)
                {
                   if(idle_time==0)
                   {
                      f_flg=1;
                      if_flg   = 0;
                      r_flg    = 0;
                      ir_flg   = 0;
                      misec    = 0;
                      idle_time=didle_time;
                   }
                   if(ir_flg==1)
                   {
                      idle_time=idle_time-1;
                      forward = 0;
                      reverse = 0;
                   }
                }
            }
            TR1=1;
            TF1=0;
            TR0=1;

      }
void adc()
{
    ALE=1;                    //A high to low pulse of ALE
    for(i=1;i<=25;)
    {i++;}

    SOC=1;                    //A high to low pulse of SOC
    for(i=1;i<=25;)
    {i++;}

    ALE=0;
    for(i=1;i<=25;)
    {i++;}

    SOC=0;
    for(i=1;i<=250;)
    {i++;}
    temperature=ADC_PORT;

}
```

CONCLUSION

6 Conclusion

This project titled "MICROCONTROLLER BASED WASHING MACHINE CONTROL THROUGH PC" has been designed, fabricated and tested successfully.

The data sheets regarding the microcontroller and all other chips used in implementing the hardware have been enclosed in the appendix.

This project is aimed at controlling two washing machines, but up to 32 machines can be controlled simultaneously in a similar way using this RS485 Communication. This full duplex automatic machine is designed mainly for the use in hospitals where a large capacity is required. The cost of such a machine is considerably reduced since individual panel control is not required. Also the error detection in any of the machines is easy since all the machines are controlled using a single PC.

Our project is basically a demonstration of an electronic concept used in the control of a washing machine, but this idea can be further extended for the implementation of other control applications

Bibliography

1. Kenneth J. Ayala, "The 8051 Microcontroller – Architecture, Programming and Applications", Penram International Publishing, 1996.
2. Douglas V. Hall, "Microprocessors and Interfacing", Tata McGraw Hill Publishing, 1999.
3. AT89C51 ATMEL Flash Microcontrollers – Manual.
4. Linear IC manual – National Semiconductor.
5. Scott Warner, "Teach Yourself Visual Basic 6.0", Tata McGraw Hill Publishing, 1999.

ures

- patible with MCS-51™ Products
- ytes of In-System Reprogrammable Flash Memory
- Endurance: 1,000 Write/Erase Cycles
- Static Operation: 0 Hz to 24 MHz
- e-level Program Memory Lock
- x 8-bit Internal RAM
- rogrammable I/O Lines
- 16-bit Timer/Counters
- nterrupt Sources
- grammable Serial Channel
- -power Idle and Power-down Modes

cription

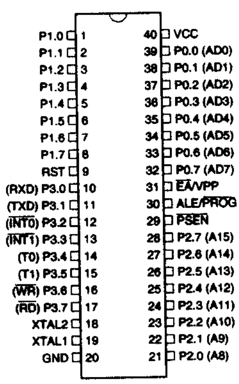
T89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K of Flash programmable and erasable read only memory (PEROM). The device nufactured using Atmel's high-density nonvolatile memory technology and is atible with the industry-standard MCS-51 instruction set and pinout. The on-chip allows the program memory to be reprogrammed in-system or by a conven- nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides nly-flexible and cost-effective solution to many embedded control applications.

8-bit Microcontroller with 4K Bytes Flash

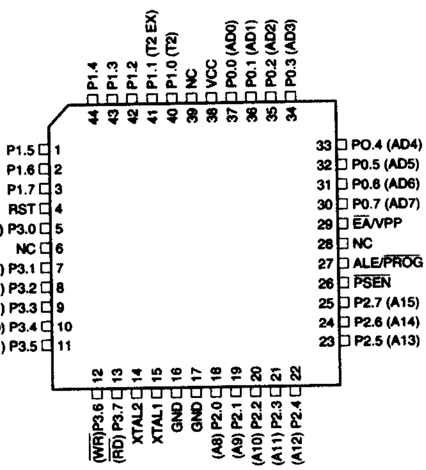
AT89C51

Configurations

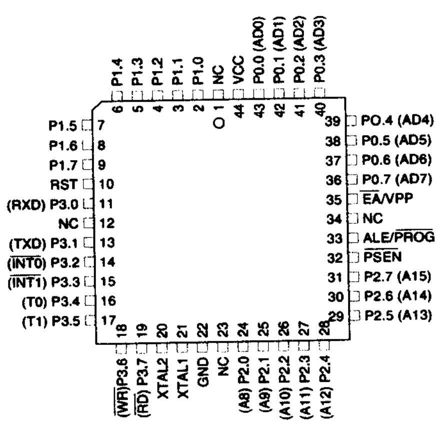
PDIP



PQFP/TQFP

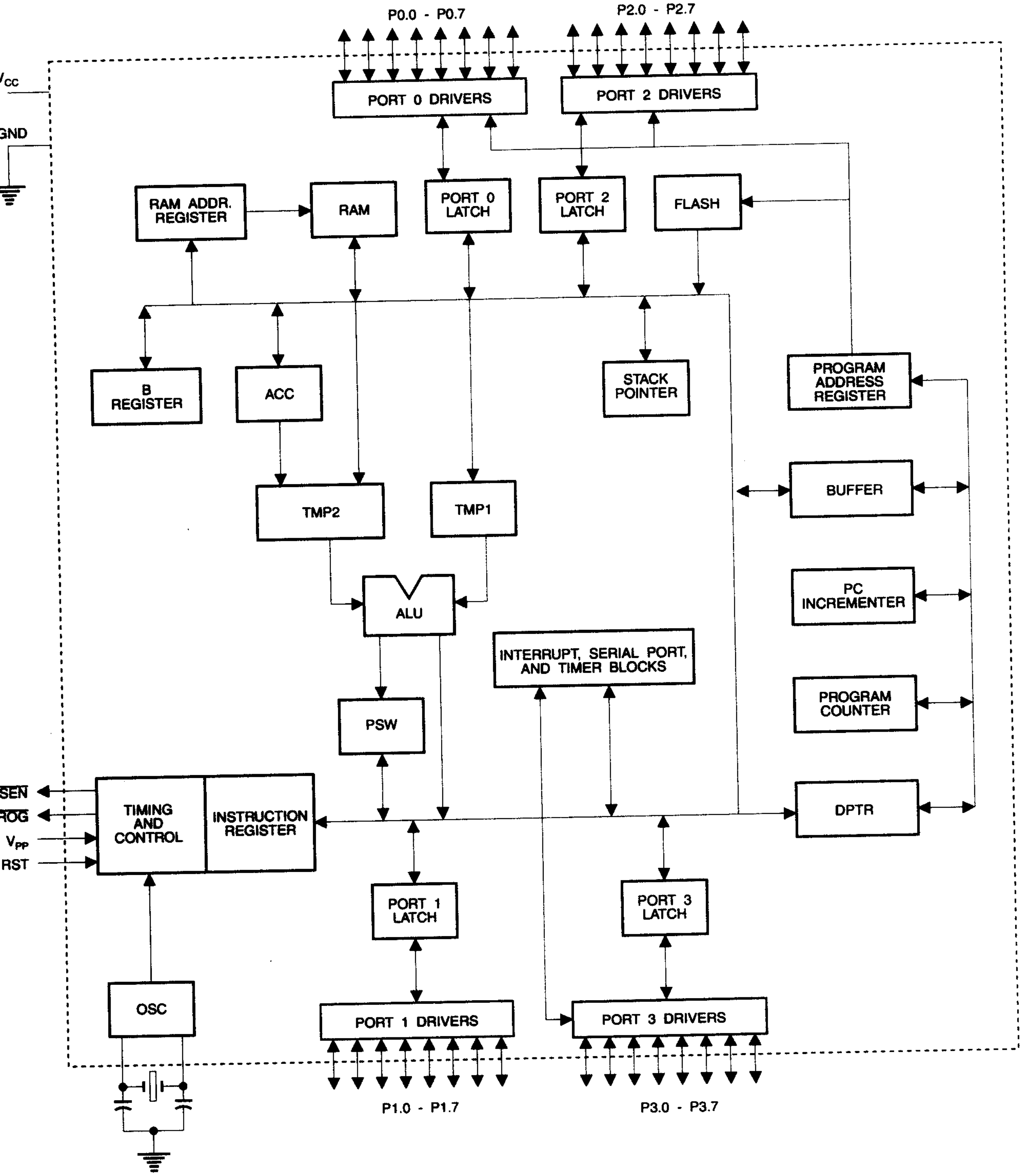


PLCC



Rev. 0265G–02/00

k Diagram



P0.0 - P0.7
P2.0 - P2.7
PORT 0 DRIVERS
PORT 2 DRIVERS
RAM ADDR. REGISTER
RAM
PORT 0 LATCH
PORT 2 LATCH
FLASH
B REGISTER
ACC
STACK POINTER
PROGRAM ADDRESS REGISTER
BUFFER
TMP2
TMP1
ALU
PC INCREMENTER
INTERRUPT, SERIAL PORT, AND TIMER BLOCKS
PSW
PROGRAM COUNTER
TIMING AND CONTROL
INSTRUCTION REGISTER
DPTR
RST
PORT 1 LATCH
PORT 3 LATCH
OSC
PORT 1 DRIVERS
PORT 3 DRIVERS
P1.0 - P1.7
P3.0 - P3.7

T89C51 provides the following standard features: 4K of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit counters, a five vector two-level interrupt architecture, duplex serial port, on-chip oscillator and clock cir- In addition, the AT89C51 is designed with static logic eration down to zero frequency and supports two re selectable power saving modes. The Idle Mode the CPU while allowing the RAM, timer/counters, port and interrupt system to continue functioning. The r-down Mode saves the RAM contents but freezes cillator disabling all other chip functions until the next are reset.

Description

y voltage.

nd.

0

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

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

0 also receives the code bytes during Flash program- , and outputs the code bytes during program cation. External pullups are required during program cation.

1

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

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

2

2 is an 8-bit bi-directional I/O port with internal pullups. Port 2 output buffers can sink/source four TTL inputs. en 1s are written to Port 2 pins they are pulled high by 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	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

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

RST

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

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

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

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

is skipped during each access to external Data
ry.

red, ALE operation can be disabled by setting bit 0 of
ocation 8EH. With the bit set, ALE is active only dur-
MOVX or MOVC instruction. Otherwise, the pin is
y pulled high. Setting the ALE-disable bit has no
if the microcontroller is in external execution mode.

am Store Enable is the read strobe to external pro-
memory.

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

PP

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

hould be strapped to V_{CC} for internal program
utions.

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

.1

to the inverting oscillator amplifier and input to the
nal clock operating circuit.

.2

ut from the inverting oscillator amplifier.

cillator Characteristics

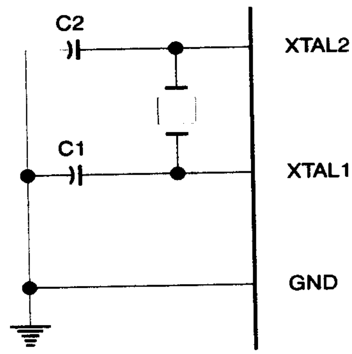
L1 and XTAL2 are the input and output, respectively,
inverting amplifier which can be configured for use as
n-chip oscillator, as shown in Figure 1. Either a quartz
tal or ceramic resonator may be used. To drive the
ce from an external clock source, XTAL2 should be left
unconnected while XTAL1 is driven 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

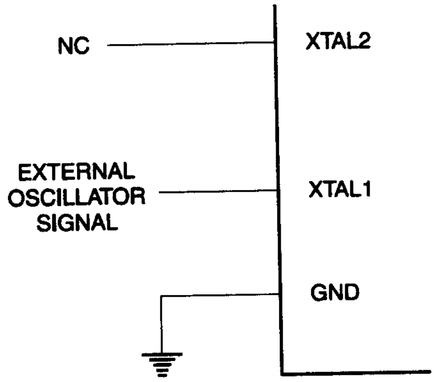


Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

tus of External Pins During Idle and Power-down Modes

de	Program Memory	ALE	$\overline{\text{PSEN}}$	PORT0	PORT1	PORT2	PORT3
	Internal	1	1	Data	Data	Data	Data
	External	1	1	Float	Data	Address	Data
wer-down	Internal	0	0	Data	Data	Data	Data
wer-down	External	0	0	Float	Data	Data	Data

2. External Clock Drive Configuration



er-down Mode

power-down mode, the oscillator is stopped, and the ction that invokes power-down is the last instruction ted. 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.

k 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

ıramming the Flash

T89C51 is normally shipped with the on-chip Flash y array in the erased state (that is, contents = FFH) ady to be programmed. The programming interface ts either a high-voltage (12-volt) or a low-voltage program enable signal. The low-voltage program-mode provides a convenient way to program the C51 inside the user's system, while the high-voltage amming mode is compatible with conventional third-Flash or EPROM programmers.

T89C51 is shipped with either the high-voltage or ltage programming mode enabled. The respective de marking and device signature codes are listed in lowing table.

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

T89C51 code memory array is programmed byte-by-in either programming mode. *To program any non- byte in the on-chip Flash Memory, the entire memory be erased using the Chip Erase Mode.*

ıramming Algorithm: Before programming the C51, the address, data and control signals should be p according to the Flash programming mode table and e 3 and Figure 4. To program the AT89C51, take the ving steps.

nput the desired memory location on the address nes.

nput the appropriate data byte on the data lines.

ctivate the correct combination of control signals.

Raise $\overline{EA}/V_{PP}$ to 12V for the high-voltage program-ming mode.

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.

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

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

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/$\overline{\text{PROG}}$ 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.

h Programming Modes

		RST	PSEN	ALE/PROG	EA/V_{PP}	P2.6	P2.7	P3.6	P3.7
Code Data		H	L	‾_/‾	H/12V	L	H	H	H
Code Data		H	L	H	H	L	L	H	H
Lock	Bit - 1	H	L	‾_/‾	H/12V	H	H	H	H
	Bit - 2	H	L	‾_/‾	H/12V	H	H	L	L
	Bit - 3	H	L	‾_/‾	H/12V	H	L	H	L
Erase		H	L	‾_/‾ (1)	H/12V	H	L	L	L
d Signature Byte		H	L	H	H	L	L	L	L

1. Chip Erase requires a 10 ms PROG pulse.

re 3. Programming the Flash

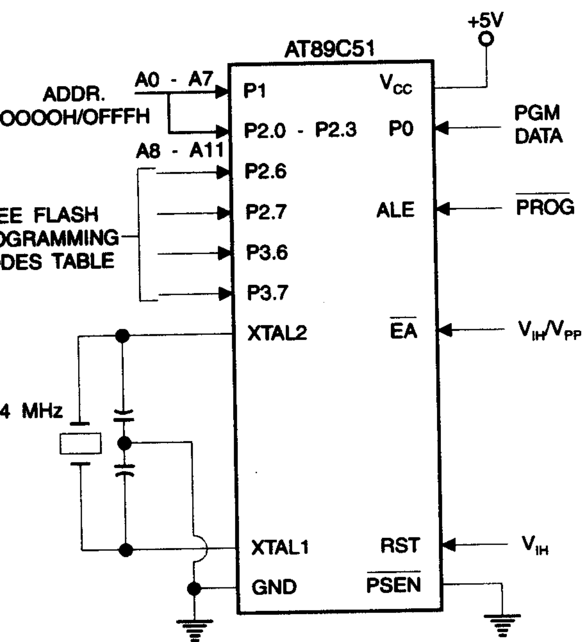
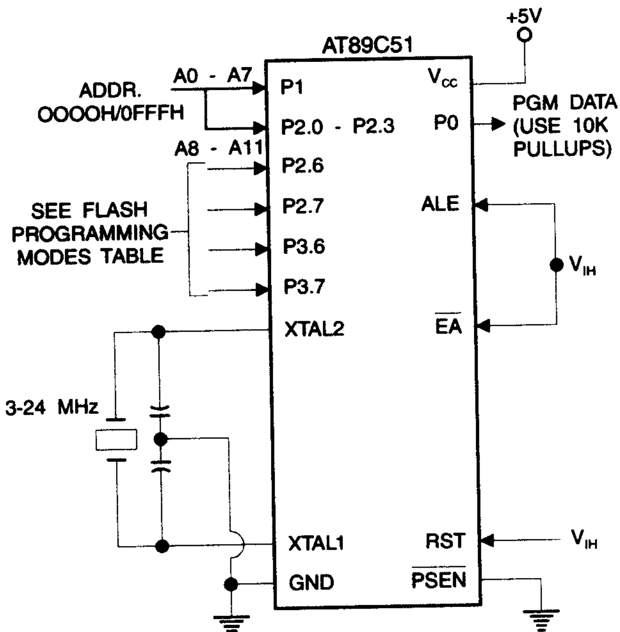
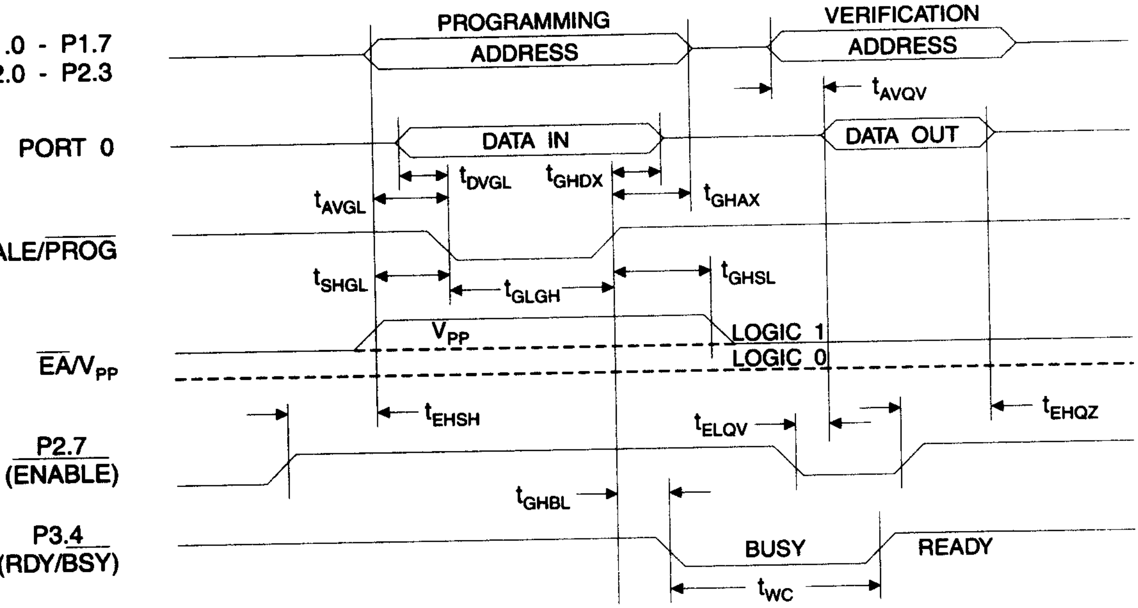


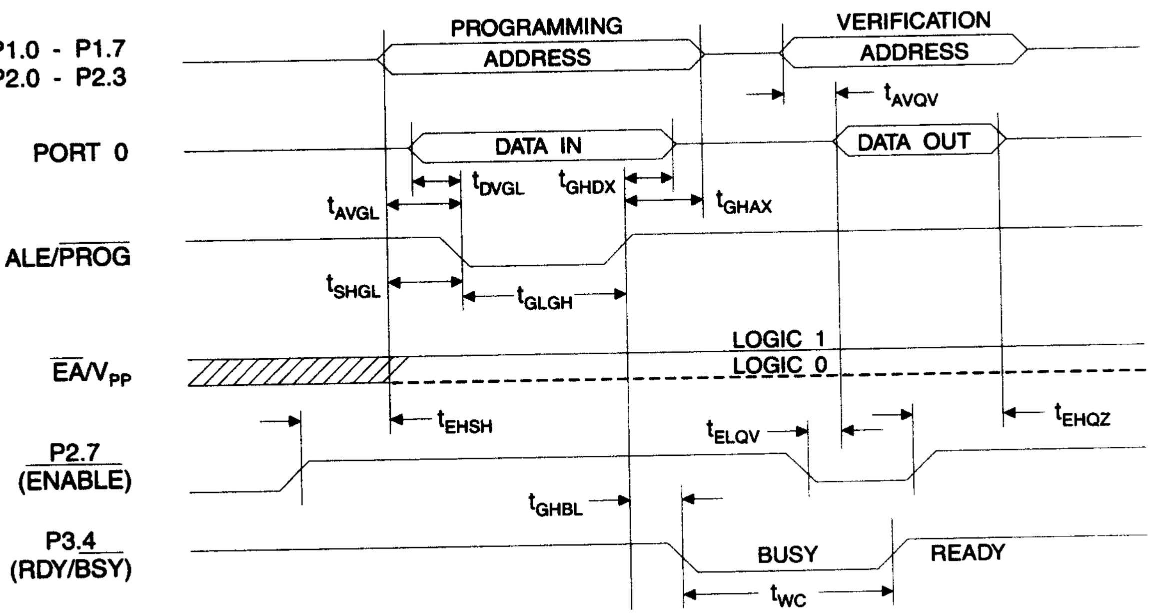
Figure 4. Verifying the Flash



h Programming and Verification Waveforms - High-voltage Mode (V_{PP} = 12V)



sh Programming and Verification Waveforms - Low-voltage Mode (V_{PP} = 5V)



h Programming and Verification Characteristics

°C to 70°C, V_{CC} = 5.0 ± 10%

ol	Parameter	Min	Max	Units
	Programming Enable Voltage	11.5	12.5	V
	Programming Enable Current		1.0	mA
L	Oscillator Frequency	3	24	MHz
	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
	Address Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
	Data Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
	P2.7 ($\overline{\text{ENABLE}}$) High to V_{PP}	$48t_{CLCL}$		
	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
(1)	V_{PP} Hold After $\overline{\text{PROG}}$	10		μs
	$\overline{\text{PROG}}$ Width	1	110	μs
	Address to Data Valid		$48t_{CLCL}$	
	$\overline{\text{ENABLE}}$ Low to Data Valid		$48t_{CLCL}$	
Z	Data Float After $\overline{\text{ENABLE}}$	0	$48t_{CLCL}$	
	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	μs
	Byte Write Cycle Time		2.0	ms

1. Only used in 12-volt programming mode.

Detailed Description

e MAX220–MAX249 contain four sections: dual arge-pump DC-DC voltage converters, RS-232 dri-'s, RS-232 receivers, and receiver and transmitter able control inputs.

Dual Charge-Pump Voltage Converter

e MAX220–MAX249 have two internal charge-pumps it convert +5V to ±10V (unloaded) for RS-232 driver eration. The first converter uses capacitor C1 to dou-e the +5V input to +10V on C3 at the V+ output. The cond converter uses capacitor C2 to invert +10V to)V on C4 at the V- output.

small amount of power may be drawn from the +10V +) and -10V (V-) outputs to power external circuitry ee the *Typical Operating Characteristics* section), cept on the MAX225 and MAX245–MAX247, where ese pins are not available. V+ and V- are not regulated, the output voltage drops with increasing load current. o not load V+ and V- to a point that violates the mini-um ±5V EIA/TIA-232E driver output voltage when urcing current from V+ and V- to external circuitry.

hen using the shutdown feature in the MAX222, AX225, MAX230, MAX235, MAX236, MAX240, AX241, and MAX245–MAX249, avoid using V+ and V- power external circuitry. When these parts are shut own, V- falls to 0V, and V+ falls to +5V. For applica-ons where a +10V external supply is applied to the V+ in (instead of using the internal charge pump to gen-rate +10V), the C1 capacitor must not be installed and e $\overline{SHDN}$ pin must be tied to V_{CC}. This is because V+ internally connected to V_{CC} in shutdown mode.

RS-232 Drivers

he typical driver output voltage swing is ±8V when aded with a nominal 5kΩ RS-232 receiver and V_{CC} = 5V. Output swing is guaranteed to meet the EIA/TIA-32E and V.28 specification, which calls for ±5V mini-ium driver output levels under worst-case conditions. hese include a minimum 3kΩ load, V_{CC} = +4.5V, and aximum operating temperature. Unloaded driver out-ut voltage ranges from (V+ -1.3V) to (V- +0.5V).

nput thresholds are both TTL and CMOS compatible. he inputs of unused drivers can be left unconnected ince 400kΩ input pull-up resistors to V_{CC} are built in except for the MAX220). The pull-up resistors force the utputs of unused drivers low because all drivers invert. he internal input pull-up resistors typically source 12µA, except in shutdown mode where the pull-ups are dis-abled. Driver outputs turn off and enter a high-imped-ance state—where leakage current is typically microamperes (maximum 25µA)—when in shutdown mode, in three-state mode, or when device power is removed. Outputs can be driven to ±15V. The power-supply current typically drops to 8µA in shutdown mode. The MAX220 does not have pull-up resistors to force the outputs of the unused drivers low. Connect unused inputs to GND or V_{CC}.

The MAX239 has a receiver three-state control line, and the MAX223, MAX225, MAX235, MAX236, MAX240, and MAX241 have both a receiver three-state control line and a low-power shutdown control. Table 2 shows the effects of the shutdown control and receiver three-state control on the receiver outputs.

The receiver TTL/CMOS outputs are in a high-imped-ance, three-state mode whenever the three-state enable line is high (for the MAX225/MAX235/MAX236/MAX239–MAX241), and are also high-impedance whenever the shutdown control line is high.

When in low-power shutdown mode, the driver outputs are turned off and their leakage current is less than 1µA with the driver output pulled to ground. The driver output leakage remains less than 1µA, even if the transmitter output is backdriven between 0V and (V_{CC} + 6V). Below -0.5V, the transmitter is diode clamped to ground with 1kΩ series impedance. The transmitter is also zener clamped to approximately V_{CC} + 6V, with a series impedance of 1kΩ.

The driver output slew rate is limited to less than 30V/µs as required by the EIA/TIA-232E and V.28 specifica-tions. Typical slew rates are 24V/µs unloaded and 10V/µs loaded with 3Ω and 2500pF.

RS-232 Receivers

EIA/TIA-232E and V.28 specifications define a voltage level greater than 3V as a logic 0, so all receivers invert. Input thresholds are set at 0.8V and 2.4V, so receivers respond to TTL level inputs as well as EIA/TIA-232E and V.28 levels.

The receiver inputs withstand an input overvoltage up to ±25V and provide input terminating resistors with

Table 2. Three-State Control of Receivers

PART	SHDN	$\overline{SHDN}$	EN	$\overline{EN(R)}$	RECEIVERS
MAX223	—	Low High High	X Low High	—	High Impedance Active High Impedance
MAX225	—	—	—	Low High	High Impedance Active
MAX235 MAX236 MAX240	Low Low High	—	—	Low High X	High Impedance Active High Impedance

minal 5kΩ values. The receivers implement Type 1 erpretation of the fault conditions of V.28 and VTIA-232E.

e receiver input hysteresis is typically 0.5V with a aranteed minimum of 0.2V. This produces clear out- t transitions with slow-moving input signals, even th moderate amounts of noise and ringing. The ceiver propagation delay is typically 600ns and is dependent of input swing direction.

Low-Power Receive Mode

ie low-power receive-mode feature of the MAX223, AX242, and MAX245–MAX249 puts the IC into shut- own mode but still allows it to receive information. This important for applications where systems are periodi- ally awakened to look for activity. Using low-power ceive mode, the system can still receive a signal that ill activate it on command and prepare it for communi- ation at faster data rates. This operation conserves ystem power.

Negative Threshold—MAX243

ne MAX243 is pin compatible with the MAX232A, differ- g only in that RS-232 cable fault protection is removed n one of the two receiver inputs. This means that control nes such as CTS and RTS can either be driven or left oating without interrupting communication. Different ables are not needed to interface with different pieces of quipment.

he input threshold of the receiver without cable fault rotection is -0.8V rather than +1.4V. Its output goes ositive only if the input is connected to a control line at is actively driven negative. If not driven, it defaults o the 0 or "OK to send" state. Normally, the MAX243's ther receiver (+1.4V threshold) is used for the data line TD or RD), while the negative threshold receiver is con- ected to the control line (DTR, DTS, CTS, RTS, etc.).

)ther members of the RS-232 family implement the ptional cable fault protection as specified by EIA/TIA- 32E specifications. This means a receiver output goes igh whenever its input is driven negative, left floating, r shorted to ground. The high output tells the serial ommunications IC to stop sending data. To avoid this, he control lines must either be driven or connected vith jumpers to an appropriate positive voltage level.

Shutdown—MAX222–MAX242

On the MAX222, MAX235, MAX236, MAX240, and MAX241, all receivers are disabled during shutdown. On the MAX223 and MAX242, two receivers continue to operate in a reduced power mode when the chip is in shutdown. Under these conditions, the propagation delay increases to about 2.5μs for a high-to-low input transition. When in shutdown, the receiver acts as a CMOS inverter with no hysteresis. The MAX223 and MAX242 also have a receiver output enable input ($\overline{EN}$ for the MAX242 and EN for the MAX223) that allows receiver output control independent of $\overline{SHDN}$ (SHDN for MAX241). With all other devices, $\overline{SHDN}$ (SHDN for MAX241) also disables the receiver outputs.

The MAX225 provides five transmitters and five receivers, while the MAX245 provides ten receivers and eight transmitters. Both devices have separate receiver and transmitter-enable controls. The charge pumps turn off and the devices shut down when a logic high is applied to the ENT input. In this state, the supply current drops to less than 25μA and the receivers continue to operate in a low-power receive mode. Driver outputs enter a high-impedance state (three-state mode). On the MAX225, all five receivers are controlled by the $\overline{ENR}$ input. On the MAX245, eight of the receiver outputs are controlled by the $\overline{ENR}$ input, while the remaining two receivers (RA5 and RB5) are always active. RA1–RA4 and RB1–RB4 are put in a three-state mode when $\overline{ENR}$ is a logic high.

Receiver and Transmitter Enable Control Inputs

The MAX225 and MAX245–MAX249 feature transmitter and receiver enable controls.

The receivers have three modes of operation: full-speed receive (normal active), three-state (disabled), and low-power receive (enabled receivers continue to function at lower data rates). The receiver enable inputs control the full-speed receive and three-state modes. The transmitters have two modes of operation: full-speed transmit (normal active) and three-state (disabled). The transmitter enable inputs also control the shutdown mode. The device enters shutdown mode when all transmitters are disabled. Enabled receivers function in the low-power receive mode when in shutdown.

oles 1a–1d define the control states. The MAX244 s no control pins and is not included in these tables.

e MAX246 has ten receivers and eight drivers with o control pins, each controlling one side of the vice. A logic high at the A-side control input ($\overline{ENA}$) uses the four A-side receivers and drivers to go into three-state mode. Similarly, the B-side control input $\overline{NB}$) causes the four B-side drivers and receivers to into a three-state mode. As in the MAX245, one A-le and one B-side receiver (RA5 and RB5) remain tive at all times. The entire device is put into shut-wn mode when both the A and B sides are disabled $\overline{NA}$ = $\overline{ENB}$ = +5V).

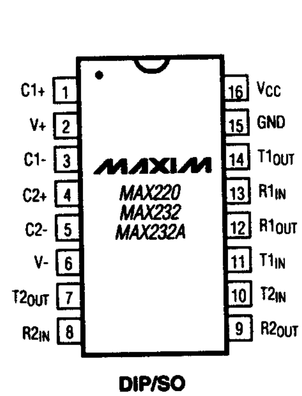
ie MAX247 provides nine receivers and eight drivers th four control pins. The $\overline{ENRA}$ and $\overline{ENRB}$ receiver able inputs each control four receiver outputs. The $\overline{ITA}$ and $\overline{ENTB}$ transmitter enable inputs each control ur drivers. The ninth receiver (RB5) is always active. ie device enters shutdown mode with a logic high on oth $\overline{ENTA}$ and $\overline{ENTB}$.

ie MAX248 provides eight receivers and eight drivers ith four control pins. The $\overline{ENRA}$ and $\overline{ENRB}$ receiver iable inputs each control four receiver outputs. The $\overline{ITA}$ and $\overline{ENTB}$ transmitter enable inputs control four ivers each. This part does not have an always-active ceiver. The device enters shutdown mode and trans-iitters go into a three-state mode with a logic high on oth $\overline{ENTA}$ and $\overline{ENTB}$.

The MAX249 provides ten receivers and six drivers with four control pins. The $\overline{ENRA}$ and $\overline{ENRB}$ receiver enable inputs each control five receiver outputs. The $\overline{ENTA}$ and $\overline{ENTB}$ transmitter enable inputs control three drivers each. There is no always-active receiver. The device enters shutdown mode and transmitters go into a three-state mode with a logic high on both $\overline{ENTA}$ and $\overline{ENTB}$. In shutdown mode, active receivers operate in a low-power receive mode at data rates up to 20kbits/sec.

Applications Information

Figures 5 through 25 show pin configurations and typical operating circuits. In applications that are sensitive to power-supply noise, V_{CC} should be decoupled to ground with a capacitor of the same value as C1 and C2 connected as close as possible to the device.



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

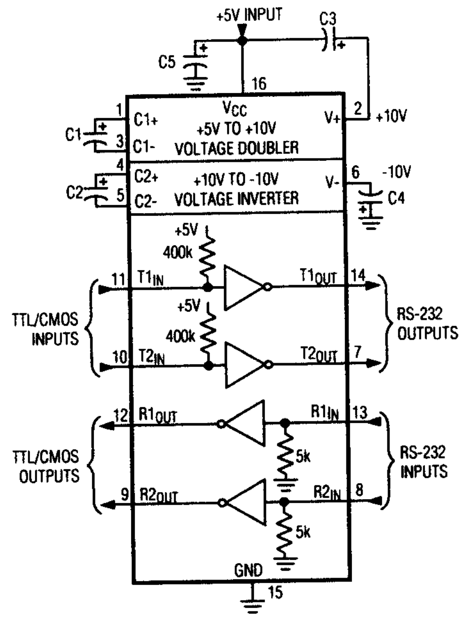
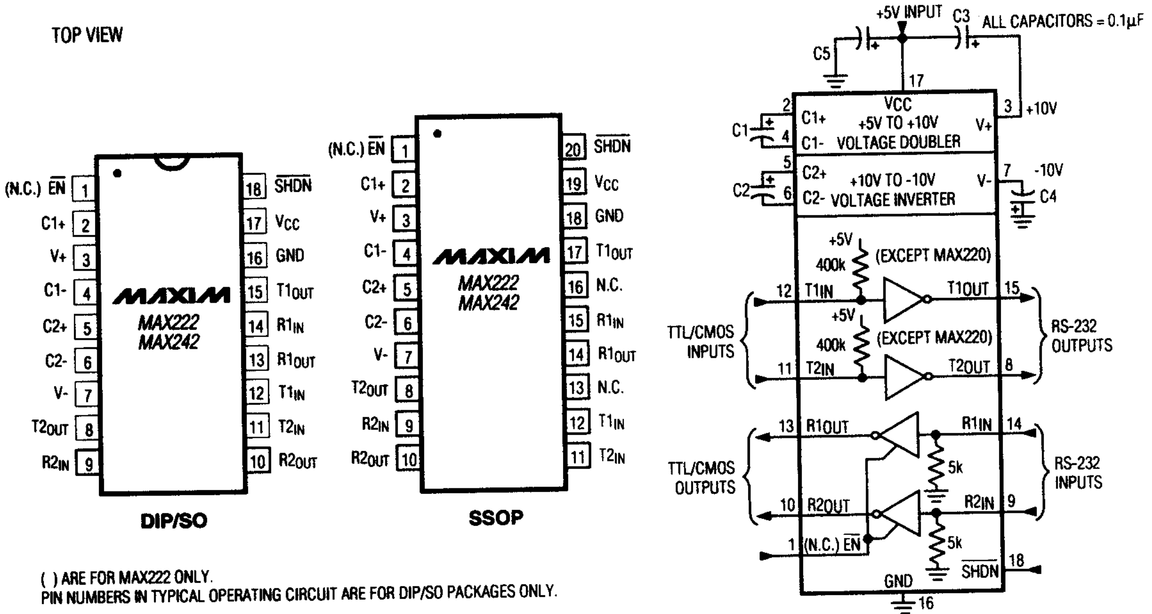


Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit



( ) ARE FOR MAX222 ONLY.
PIN NUMBERS IN TYPICAL OPERATING CIRCUIT ARE FOR DIP/SO PACKAGES ONLY.

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

July 1998

National Semiconductor

DS75176B/DS75176BT Multipoint RS-485/RS-422 Transceivers

General Description

The DS75176B is a high speed differential TRI-STATE® bus/line transceiver designed to meet the requirements of EIA standard RS485 with extended common mode range (+12V to −7V), for multipoint data transmission. In addition, it is compatible with RS-422.

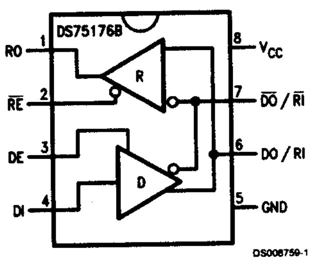
The driver and receiver outputs feature TRI-STATE capability, for the driver outputs over the entire common mode range of +12V to −7V. Bus contention or fault situations that cause excessive power dissipation within the device are handled by a thermal shutdown circuit, which forces the driver outputs into the high impedance state.

DC specifications are guaranteed over the 0 to 70°C temperature and 4.75V to 5.25V supply voltage range.

Features

- Meets EIA standard RS485 for multipoint bus transmission and is compatible with RS-422.
- Small Outline (SO) Package option available for minimum board space.
- 22 ns driver propagation delays.
- Single +5V supply.
- −7V to +12V bus common mode range permits ±7V ground difference between devices on the bus.
- Thermal shutdown protection.
- High impedance to bus with driver in TRI-STATE or with power off, over the entire common mode range allows the unused devices on the bus to be powered down.
- Pin out compatible with DS3695/A and SN75176A/B.
- Combined impedance of a driver output and receiver input is less than one RS485 unit load, allowing up to 32 transceivers on the bus.
- 70 mV typical receiver hysteresis.

Connection and Logic Diagram



Top View
Order Number DS75176BN, DS75176BTN, DS75176BM or DS75176BTM
See NS Package Number N08E or M08A

Absolute Maximum Ratings (Note 1)

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

Supply Voltage, V_{CC}	7V
Control Input Voltages	7V
Driver Input Voltage	7V
Driver Output Voltages	+15V/ −10V
Receiver Input Voltages (DS75176B)	+15V/ −10V
Receiver Output Voltage	5.5V
Continuous Power Dissipation @ 25°C	
for M Package	675 mW (Note 5)
for N Package	900 mW (Note 4)
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 4 seconds)	260°C

Recommended Operating Conditions

	Min	Max	Units
Supply Voltage, V_{CC}	4.75	5.25	V
Voltage at Any Bus Terminal (Separate or Common Mode)	−7	+12	V
Operating Free Air Temperature T_A			
DS75176B	0	+70	°C
DS75176BT	−40	+85	°C
Differential Input Voltage, VID (Note 6)	−12	+12	V

Electrical Characteristics (Notes 2, 3)

$0°C \le T_A \le 70°C$, $4.75V < V_{CC} < 5.25V$ unless otherwise specified

Symbol	Parameter		Conditions		Min	Typ	Max	Units
V_{OD1}	Differential Driver Output Voltage (Unloaded)		$I_O = 0$				5	V
V_{OD2}	Differential Driver Output Voltage (with Load)		(*Figure 1*)	R = 50Ω; (RS-422) (Note 7)	2			V
				R = 27Ω; (RS-485)	1.5			V
ΔV_{OD}	Change in Magnitude of Driver Differential Output Voltage For Complementary Output States		(*Figure 1*)	R = 27Ω			0.2	V
V_{OC}	Driver Common Mode Output Voltage		(*Figure 1*)	R = 27Ω			3.0	V
$\Delta\lvert V_{OC}\rvert$	Change in Magnitude of Driver Common Mode Output Voltage For Complementary Output States		(*Figure 1*)	R = 27Ω			0.2	V
V_{IH}	Input High Voltage	DI, DE, $\overline{RE}$, E			2			V
V_{IL}	Input Low Voltage	DI, DE, $\overline{RE}$, E					0.8	
V_{CL}	Input Clamp Voltage	DI, DE, $\overline{RE}$, E		$I_{IN} = -18$ mA			−1.5	
I_{IL}	Input Low Current	DI, DE, $\overline{RE}$, E		$V_{IL} = 0.4V$			−200	μA
I_{IH}	Input High Current	DI, DE, $\overline{RE}$, E		$V_{IH} = 2.4V$			20	μA
I_{IN}	Input Current	DO/RI, $\overline{DO/RI}$	V_{CC} = 0V or 5.25V DE = 0V	$V_{IN} = 12V$			+1.0	mA
				$V_{IN} = -7V$			−0.8	mA
V_{TH}	Differential Input Threshold Voltage for Receiver		$-7V \le V_{CM} \le +12V$		−0.2		+0.2	V
ΔV_{TH}	Receiver Input Hysteresis		$V_{CM} = 0V$			70		mV
V_{OH}	Receiver Output High Voltage		$I_{OH} = -400$ μA		2.7			V
V_{OL}	Output Low Voltage	RO	I_{OL} = 16 mA (Note 7)				0.5	V
I_{OZR}	OFF-State (High Impedance) Output Current at Receiver		V_{CC} = Max, $0.4V \le V_O \le 2.4V$				±20	μA
R_{IN}	Receiver Input Resistance		$-7V \le V_{CM} \le +12V$		12			kΩ
I_{CC}	Supply Current		No Load (Note 7)	Driver Outputs Enabled			55	mA
				Driver Outputs Disabled			35	mA
I_{OSD}	Driver Short-Circuit Output Current		$V_O = -7V$ (Note 7)				−250	mA
			$V_O = +12V$ (Note 7)				+250	mA

Electrical Characteristics (Notes 2, 3) (Continued)

0°C ≤ T_A ≤ 70°C, 4.75V < V_{CC} < 5.25V unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I_{OSR}	Receiver Short-Circuit Output Current	V_O = 0V	−15		−85	mA

Note 1: "Absolute Maximum Ratings" are those beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the device should be operated at these limits. The tables of "Electrical Characteristics" provide conditions for actual device operation.

Note 2: All currents into device pins are positive; all currents out of device pins are negative. All voltages are referenced to device ground unless otherwise specified.

Note 3: All typicals are given for V_{CC} = 5V and T_A = 25°C.

Note 4: Derate linearly at 5.56 mW/°C to 650 mW at 70°C.

Note 5: Derate linearly 6.11 mW/°C to 400 mW at 70°C.

Note 6: Differential - Input/Output bus voltage is measured at the noninverting terminal A with respect to the inverting terminal B.

Note 7: All worst case parameters for which note 7 is applied, must be increased by 10% for DS75176BT. The other parameters remain valid for −40°C < T_A < +85°C.

Switching Characteristics

V_{CC} = 5.0V, T_A = 25°C

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{PLH}	Driver Input to Output	R_{LDIFF} = 60Ω		12	22	ns
t_{PHL}	Driver Input to Output	C_{L1} = C_{L2} = 100 pF		17	22	ns
t_r	Driver Rise Time	R_{LDIFF} = 60Ω			18	ns
t_f	Driver Fall Time	C_{L1} = C_{L2} = 100 pF (*Figure 3* and *Figure 5*)			18	ns
t_{ZH}	Driver Enable to Output High	C_L = 100 pF (*Figure 4* and *Figure 6*) S1 Open		29	100	ns
t_{ZL}	Driver Enable to Output Low	C_L = 100 pF (*Figure 4* and *Figure 6*) S2 Open		31	60	ns
t_{LZ}	Driver Disable Time from Low	C_L = 15 pF (*Figure 4* and *Figure 6*) S2 Open		13	30	ns
t_{HZ}	Driver Disable Time from High	C_L = 15 pF (*Figure 4* and *Figure 6*) S1 Open		19	200	ns
t_{PLH}	Receiver Input to Output	C_L = 15 pF (*Figure 2* and *Figure 7*)		30	37	ns
t_{PHL}	Receiver Input to Output	S1 and S2 Closed		32	37	ns
t_{ZL}	Receiver Enable to Output Low	C_L = 15 pF (*Figure 2* and *Figure 8*) S2 Open		15	20	ns
t_{ZH}	Receiver Enable to Output High	C_L = 15 pF (*Figure 2* and *Figure 8*) S1 Open		11	20	ns
t_{LZ}	Receiver Disable from Low	C_L = 15 pF (*Figure 2* and *Figure 8*) S2 Open		28	32	ns
t_{HZ}	Receiver Disable from High	C_L = 15 pF (*Figure 2* and *Figure 8*) S1 Open		13	35	ns

AC Test Circuits

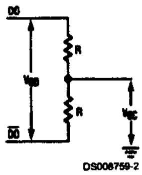
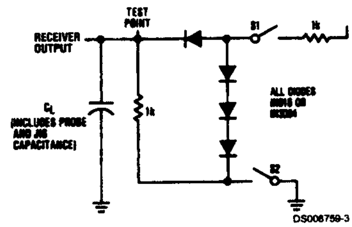


FIGURE 1.



Note: S1 and S2 of load circuit are closed except as otherwise mentioned.

FIGURE 2.

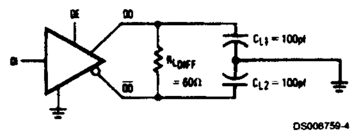
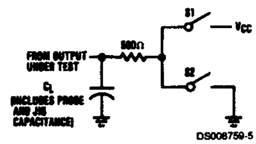


FIGURE 3.



Note: Unless otherwise specified the switches are closed.

FIGURE 4.

Switching Time Waveforms

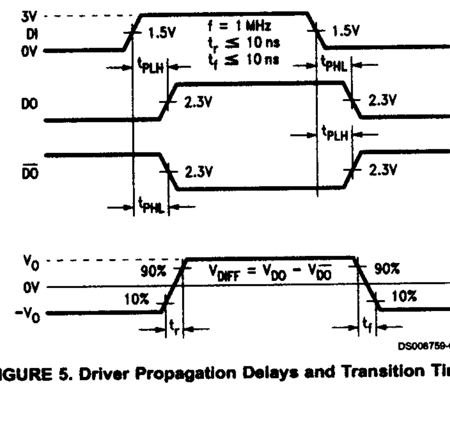


FIGURE 5. Driver Propagation Delays and Transition Times

Switching Time Waveforms (Continued)

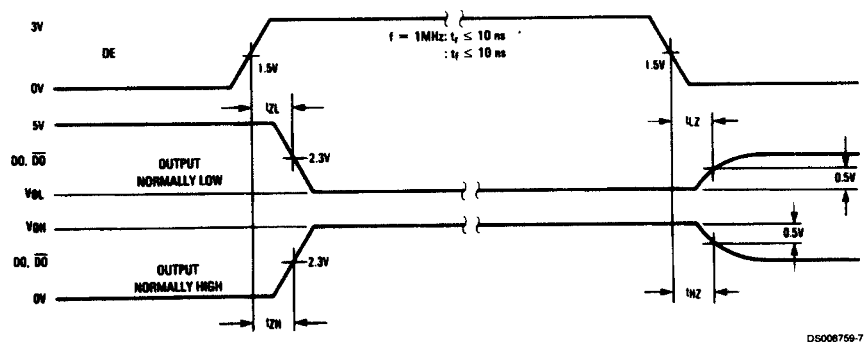
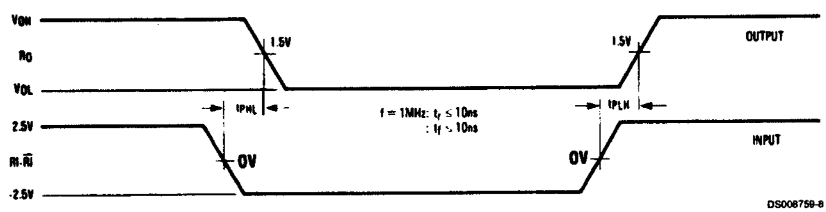


FIGURE 6. Driver Enable and Disable Times



Note: Differential input voltage may may be realized by grounding $\overline{RI}$ and pulsing RI between +2.5V and −2.5V

FIGURE 7. Receiver Propagation Delays

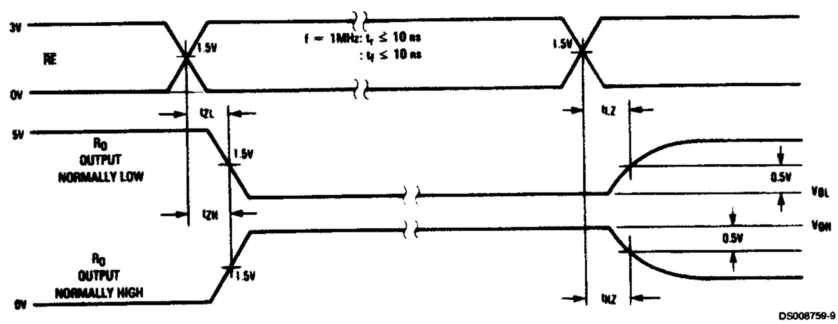


FIGURE 8. Receiver Enable and Disable Times

Function Tables

DS75176B Transmitting

Inputs			Line	Outputs	
$\overline{RE}$	DE	DI	Condition	$\overline{DO}$	DO
X	1	1	No Fault	0	1
X	1	0	No Fault	1	0
X	0	X	X	Z	Z
X	1	X	Fault	Z	Z

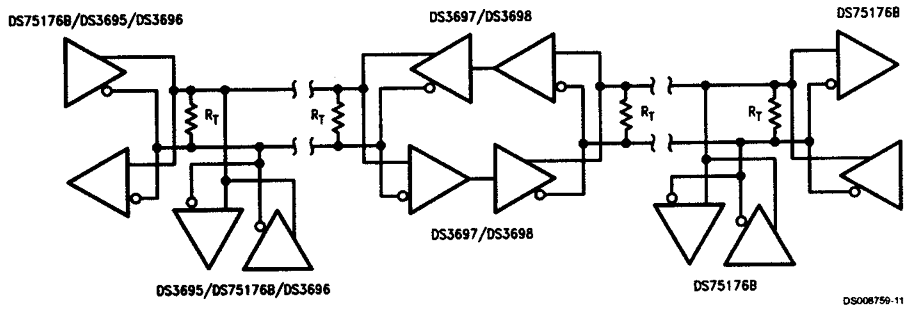
Function Tables (Continued)

DS75176B Receiving

Inputs			Outputs
$\overline{RE}$	DE	RI-$\overline{RI}$	RO
0	0	≥ +0.2V	1
0	0	≤ −0.2V	0
0	0	Inputs Open**	1
1	0	X	Z

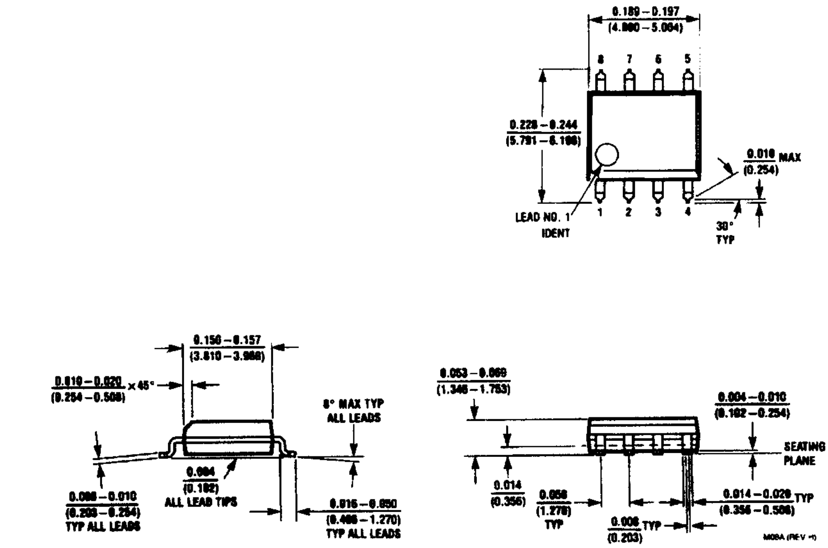
X — Don't care condition
Z — High impedance state
Fault — Improper line conditons causing excessive power dissipation in the driver, such as shorts or bus contention situations
**This is a fail safe condition

Typical Application

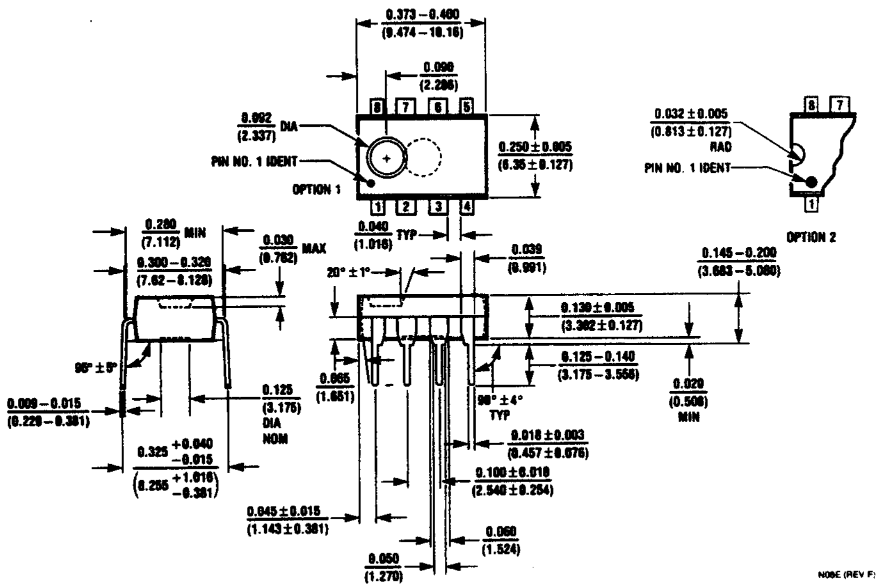


DS008759-11

Physical Dimensions inches (millimeters) unless otherwise noted



Lit. # 103669



Molded Dual-In-Line Package (N)
Order Number DS75176BN or DS75176BTN
NS Package Number N08E

October 1999

National Semiconductor

ADC0808/ADC0809
8-Bit µP Compatible A/D Converters with 8-Channel Multiplexer

General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.)

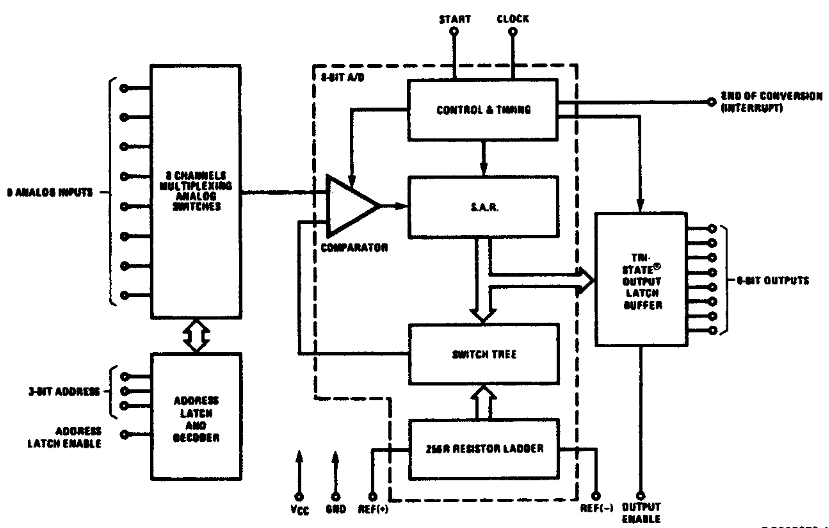
Features

- Easy interface to all microprocessors
- Operates ratiometrically or with 5 V_{DC} or analog span adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications
- Standard hermetic or molded 28-pin DIP package
- 28-pin molded chip carrier package
- ADC0808 equivalent to MM74C949
- ADC0809 equivalent to MM74C949-1

Key Specifications

- Resolution: 8 Bits
- Total Unadjusted Error: ±½ LSB and ±1 LSB
- Single Supply: 5 V_{DC}
- Low Power: 15 mW
- Conversion Time: 100 µs

Block Diagram

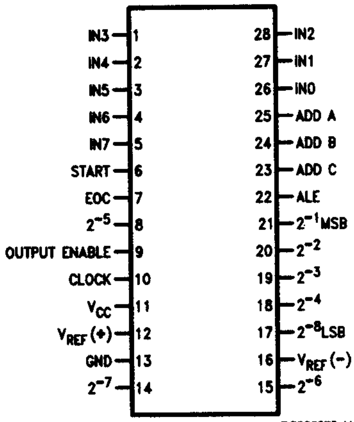


DS005672-1

See Ordering Information

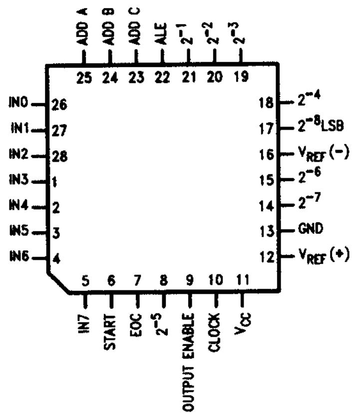
Connection Diagrams

Dual-In-Line Package



Order Number ADC0808CCN or ADC0809CCN
See NS Package J28A or N28A

Molded Chip Carrier Package



Order Number ADC0808CCV or ADC0809CCV
See NS Package V28A

Ordering Information

TEMPERATURE RANGE		−40˚C to +85˚C			−55˚C to +125˚C
Error	±½ LSB Unadjusted	ADC0808CCN	ADC0808CCV	ADC0808CCJ	ADC0808CJ
	±1 LSB Unadjusted	ADC0809CCN	ADC0809CCV		
Package Outline		N28A Molded DIP	V28A Molded Chip Carrier	J28A Ceramic DIP	J28A Ceramic DIP

bsolute Maximum Ratings (Notes 2, 1)

Military/Aerospace specified devices are required, ease contact the National Semiconductor Sales Office/ stributors for availability and specifications.

ipply Voltage (V_{CC}) (Note 3)	6.5V
ltage at Any Pin	−0.3V to (V_{CC}+0.3V)
Except Control Inputs	
ltage at Control Inputs	−0.3V to +15V
(START, OE, CLOCK, ALE, ADD A, ADD B, ADD C)	
orage Temperature Range	−65°C to +150°C
ackage Dissipation at T_A=25°C	875 mW
ad Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (plastic)	260°C
Dual-In-Line Package (ceramic)	300°C
Molded Chip Carrier Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 8)	400V

Operating Conditions (Notes 1, 2)

Temperature Range (Note 1)	$T_{MIN}≤T_A≤T_{MAX}$
ADC0808CCN,ADC0809CCN	−40°C≤T_A≤+85°C
ADC0808CCV, ADC0809CCV	−40°C ≤ T_A ≤ +85°C
Range of V_{CC} (Note 1)	4.5 V_{DC} to 6.0 V_{DC}

lectrical Characteristics

nverter Specifications: V_{CC}=5 V_{DC}=V_{REF+}, $V_{REF(-)}$=GND, $T_{MIN}≤T_A≤T_{MAX}$ and f_{CLK}=640 kHz unless otherwise stated.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	ADC0808					
	Total Unadjusted Error	25°C			±½	LSB
	(Note 5)	T_{MIN} to T_{MAX}			±¾	LSB
	ADC0809					
	Total Unadjusted Error	0°C to 70°C			±1	LSB
	(Note 5)	T_{MIN} to T_{MAX}			±1¼	LSB
	Input Resistance	From Ref(+) to Ref(−)	1.0	2.5		kΩ
	Analog Input Voltage Range	(Note 4) V(+) or V(−)	GND−0.10		V_{CC}+0.10	V_{DC}
EF(+)	Voltage, Top of Ladder	Measured at Ref(+)		V_{CC}	V_{CC}+0.1	V
$\frac{REF(+)+V_{REF(-)}}{2}$	Voltage, Center of Ladder		V_{CC}/2-0.1	V_{CC}/2	V_{CC}/2+0.1	V
EF(−)	Voltage, Bottom of Ladder	Measured at Ref(−)	−0.1	0		V
	Comparator Input Current	f_c=640 kHz, (Note 6)	−2	±0.5	2	μA

lectrical Characteristics

igital Levels and DC Specifications: ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, 4.75≤V_{CC}≤5.25V, 0°C≤T_A≤+85°C unless otherwise noted

ymbol	Parameter	Conditions	Min	Typ	Max	Units
ALOG MULTIPLEXER						
F(+)	OFF Channel Leakage Current	V_{CC}=5V, V_{IN}=5V,				
		T_A=25°C		10	200	nA
		T_{MIN} to T_{MAX}			1.0	μA
F(−)	OFF Channel Leakage Current	V_{CC}=5V, V_{IN}=0,				
		T_A=25°C	−200	−10		nA
		T_{MIN} to T_{MAX}	−1.0			μA
NTROL INPUTS						
(1)	Logical "1" Input Voltage		V_{CC}−1.5			V
(0)	Logical "0" Input Voltage				1.5	V
1)	Logical "1" Input Current (The Control Inputs)	V_{IN}=15V			1.0	μA
0)	Logical "0" Input Current (The Control Inputs)	V_{IN}=0	−1.0			μA
	Supply Current	f_{CLK}=640 kHz		0.3	3.0	mA

Electrical Characteristics (Continued)

Digital Levels and DC Specifications: ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, $4.75 \le V_{CC} \le 5.25V$, $-40^{\circ}C \le T_A \le +85^{\circ}C$ unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
DATA OUTPUTS AND EOC (INTERRUPT)						
$V_{OUT(1)}$	Logical "1" Output Voltage	V_{CC} = 4.75V I_{OUT} = −360µA I_{OUT} = −10µA		2.4 4.5		V(min) V(min)
$V_{OUT(0)}$	Logical "0" Output Voltage	I_O=1.6 mA			0.45	V
$V_{OUT(0)}$	Logical "0" Output Voltage EOC	I_O=1.2 mA			0.45	V
I_{OUT}	TRI-STATE Output Current	V_O=5V V_O=0	 −3		3	µA µA

Electrical Characteristics

Timing Specifications $V_{CC}=V_{REF(+)}=5V$, $V_{REF(-)}$=GND, $t_r=t_f$=20 ns and $T_A=25^{\circ}C$ unless otherwise noted.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{WS}	Minimum Start Pulse Width	(*Figure 5*)		100	200	ns
t_{WALE}	Minimum ALE Pulse Width	(*Figure 5*)		100	200	ns
t_s	Minimum Address Set-Up Time	(*Figure 5*)		25	50	ns
t_H	Minimum Address Hold Time	(*Figure 5*)		25	50	ns
t_D	Analog MUX Delay Time From ALE	R_S=0Ω (*Figure 5*)		1	2.5	µs
t_{H1}, t_{H0}	OE Control to Q Logic State	C_L=50 pF, R_L=10k (*Figure 8*)		125	250	ns
t_{1H}, t_{0H}	OE Control to Hi-Z	C_L=10 pF, R_L=10k (*Figure 8*)		125	250	ns
t_c	Conversion Time	f_c=640 kHz, (*Figure 5*) (Note 7)	90	100	116	µs
f_c	Clock Frequency		10	640	1280	kHz
t_{EOC}	EOC Delay Time	(*Figure 5*)	0		8+2 µS	Clock Periods
C_{IN}	Input Capacitance	At Control Inputs		10	15	pF
C_{OUT}	TRI-STATE Output Capacitance	At TRI-STATE Outputs		10	15	pF

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to GND, unless othewise specified.

Note 3: A zener diode exists, internally, from V_{CC} to GND and has a typical breakdown voltage of 7 V_{DC}.

Note 4: Two on-chip diodes are tied to each analog input which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC}n supply. The spec allows 100 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 100 mV, the output code will be correct. To achieve an absolute 0V_{DC} to 5V_{DC} input voltage range will therefore require a minimum supply voltage of 4.900 V_{DC} over temperature variations, initial tolerance and loading.

Note 5: Total unadjusted error includes offset, full-scale, linearity, and multiplexer errors. See *Figure 3*. None of these A/Ds requires a zero or full-scale adjust. However, if an all zero code is desired for an analog input other than 0.0V, or if a narrow full-scale span exists (for example: 0.5V to 4.5V full-scale) the reference voltages can be adjusted to achieve this. See *Figure 13*.

Note 6: Comparator input current is a bias current into or out of the chopper stabilized comparator. The bias current varies directly with clock frequency and has little temperature dependence (*Figure 6*). See paragraph 4.0.

Note 7: The outputs of the data register are updated one clock cycle before the rising edge of EOC.

Note 8: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

unctional Description

ıltiplexer. The device contains an 8-channel single-ended alog signal multiplexer. A particular input channel is se-cted by using the address decoder. *Table 1* shows the input ates for the address lines to select any channel. The dress is latched into the decoder on the low-to-high tran-ion of the address latch enable signal.

TABLE 1.

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

ONVERTER CHARACTERISTICS

he Converter

ne heart of this single chip data acquisition system is its -bit analog-to-digital converter. The converter is designed to ve fast, accurate, and repeatable conversions over a wide nge of temperatures. The converter is partitioned into 3 ajor sections: the 256R ladder network, the successive pproximation register, and the comparator. The converter's igital outputs are positive true.

he 256R ladder network approach (*Figure 1*) was chosen ver the conventional R/2R ladder because of its inherent onotonicity, which guarantees no missing digital codes. lonotonicity is particularly important in closed loop feedback ontrol systems. A non-monotonic relationship can cause scillations that will be catastrophic for the system. Addition-lly, the 256R network does not cause load variations on the eference voltage.

The bottom resistor and the top resistor of the ladder network in *Figure 1* are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occurs when the analog signal has reached +½ LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

The successive approximation register (SAR) performs 8 iterations to approximate the input voltage. For any SAR type converter, n-iterations are required for an n-bit converter. *Figure 2* shows a typical example of a 3-bit converter. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.

The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion (SC) pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 8 clock pulses after the rising edge of start conversion.

The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter. It is also the comparator drift which has the greatest influence on the repeatability of the device. A chopper-stabilized comparator provides the most effective method of satisfying all the converter requirements.

The chopper-stabilized comparator converts the DC input signal into an AC signal. This signal is then fed through a high gain AC amplifier and has the DC level restored. This technique limits the drift component of the amplifier since the drift is a DC component which is not passed by the AC amplifier. This makes the entire A/D converter extremely insensitive to temperature, long term drift and input offset errors.

Figure 4 shows a typical error curve for the ADC0808 as measured using the procedures outlined in AN-179.

Functional Description (Continued)

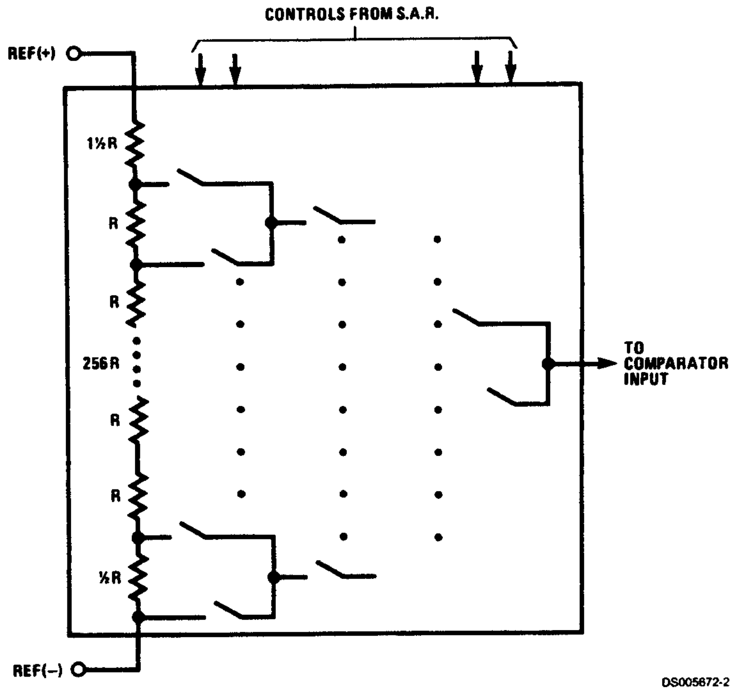


FIGURE 1. Resistor Ladder and Switch Tree

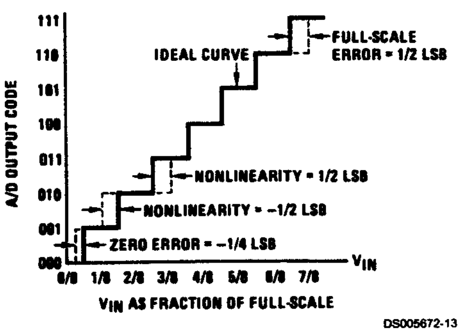


FIGURE 2. 3-Bit A/D Transfer Curve

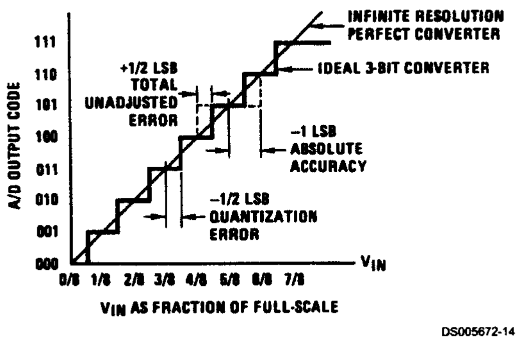


FIGURE 3. 3-Bit A/D Absolute Accuracy Curve

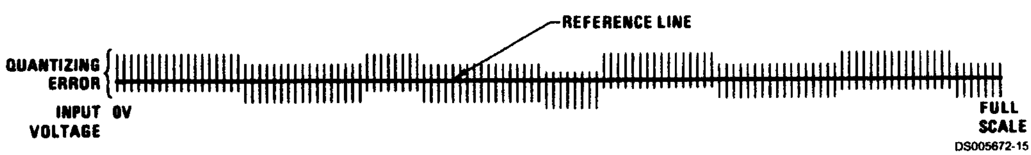


FIGURE 4. Typical Error Curve

iming Diagram

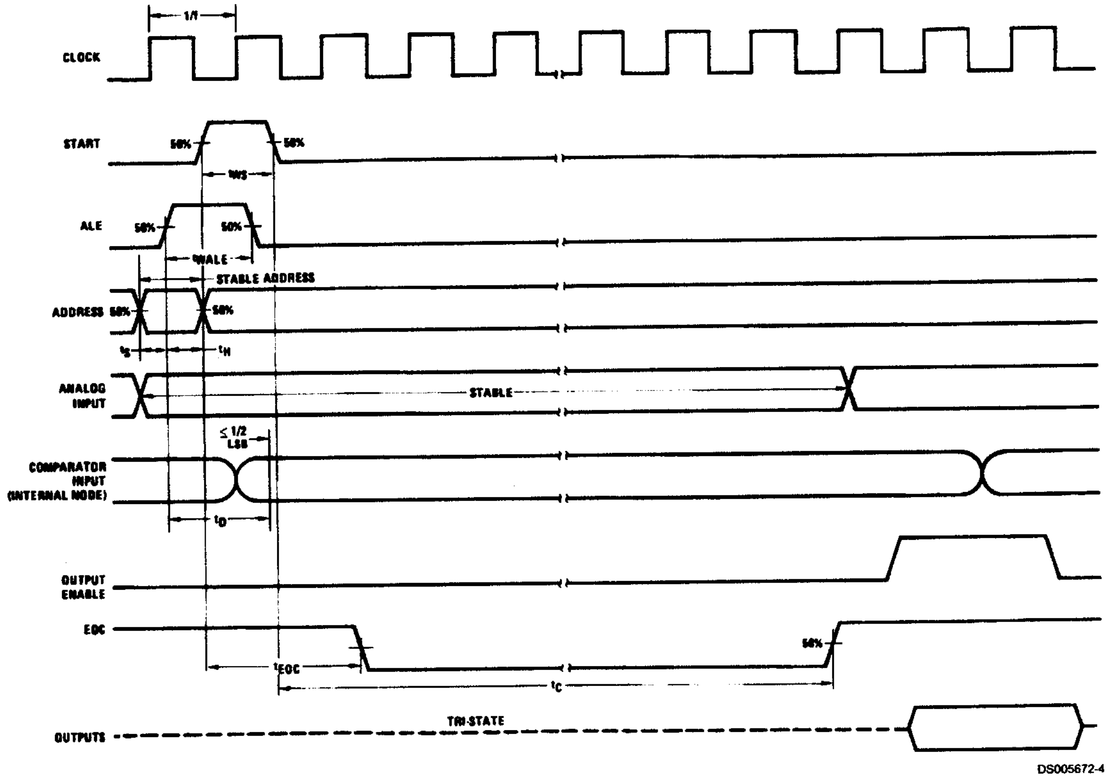


FIGURE 5.