-1984

ISO 9001:2000
Certified

WIRELESS PATIENT CARE SYSTEM

P-1180

PROJECT REPORT

SUBMITTED BY

T.RAJA-0027S0711

V.SIVANESAN-0027S0109

N.VIJAYA BASKAR-0027S0120

GUIDED BY

Mrs.N .CHITRADEVI M.E.,

(Senior Lecturer)

2004

In partial fulfillment of the requirements for the Award of the degree of ***Bachelor Of Engineering*** *in* ***Information Technology*** *Branch of* ***Bharathiar University, Coimbatore***

DEPARTMENT OF INFORMTION TECHNOLOGY

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641 006

Estd-1984

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641 006

DEPARTMENT OF INFORMATION TECHNOLOGY

ISO 9001:2000 Certified

CERTIFICATE

This is to certify that the project report entitled

WIRELESS PATIENT CARE SYSTEM

has been submitted by

T.RAJA(0027S0711)
V.SIVANESAN(0027S0109)
N.VIJAYA BASKAR(0027S0120)

In partial fulfillment of the requirements for the Award of the Degree of Bachelor of Engineering In Information Technology Branch of Bharathiar University, Coimbatore –641 046 during the Academic Year 2003-04

Head of the Department

Project Guide

Submitted for the Examination held on 26.03.2004

Internal Examiner 26/3

External Examiner 26/3/04

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

The exhilaration achieved upon the successful completion of any task should be definitely shared with the people behind the venture. This project is an amalgam of study and experience of many people without whose help this project would not have taken shape.

At the onset, we take this opportunity to thank the management of our college for having provided us excellent facilities to work with. We express our deep gratitude to our Principal **Dr.K.K.Padmanabhan B.Sc(Engg),M.Tech.**, for ushering us in the path of triumph.

We are always thankful to our beloved Professor and the Head of the Department, **Prof.S.Thangasamy B.E.(HONS).**, whose consistent support and enthusiastic involvement helped us a great deal.

We are greatly indebted to our beloved guide **Mrs.Chitradevi.N M.E.**, Senior Lecturer, Department of Computer Science and Engineering for his excellent guidance and timely support during the course of this project. As a token of our esteem and gratitude, we honour him for his assistance towards this cause.

We also thank our Project Coordinator **Mrs.S.Devaki M.E.**, and our beloved class advisor **Ms.P.Sudha B.E.**, for their invaluable assistance.

We also feel elated in manifesting our deep sense of gratitude to all the staff and lab technicians in the Department of Computer Science and Engineering.

We feel proud to pay our respectful thanks to our Parents for their enthusiasm and encouragement and also we thank our friends who have associated themselves to bring out this project successfully.

DEDICATED
TO
OUR BELOVED
PARENTS

SYNOPSIS

SYNOPSIS

In the fast moving world,the usage of wireless application is one of the indeed part of the communication system.So this type of communication is done in a effective manner with the help of FM transmitter and receiver,which is used in our project.Embedded System is one of the emerging fields in the area of communication.So we incorporate the embedded system in order to increase the stability and reliability of the system. Communication between all devices are being implemented in wireless for the advantage it possess over conventional wired data communication ,that it increases the mobility of the system and many more.

Our project aims at monitoring body condition of a patient. Using sensors, the body conditions are monitored and send to the system through com port serially. If any abnormality is sensed the microcontroller in the transmitter side sends an alert signal to the receiver side which in turn is fm modulated and received by the receiver. This causes the beeper to start beeping and alerts the main doctor.....

CONTENTS

CONTENTS

INTRODUCTION

1. INTRODUCTION

1.1 EXISTING SYSTEMS AND LIMITATIONS

In the existing systems ,the patients in the intensive care unit has to be monitored regularly because at any time his body condition may go abnormal, but in big hospitals it is impossible for a chief doctor to constantly monitor the patients in intensive care unit as there may be two or three intensive care units in the hospital.

1.2 PROPOSED SYSTEMS AND ADVANTAGES

In the proposed system, we use sensors which measure the body condition of patient and it constantly keep sending the body condition of the patient for every 5 seconds. If any abnormal condition is detected it gives an alert signal to chief doctor through wireless medium, which makes doctors task simple.

1.3 INTRODUCTION TO EMBEDDED SYSTEM

An Embedded Product uses a microchip (or) micro controller to do a task In an Embedded system there is only one application software that is typically burned into ROM. In this Project PIC-16f84A Microcontroller, operates with 8 bit data.

The embedded system comprises variety of hardware and software components, which perform specific function in the host system. It is designed around a microcontroller unit, which integrates on-chip program memory, on-chip oscillator, interrupt, port.

1.4 COMPUTER COMMUNICATION

This term refers to sending, receiving and processing information by electric means. Depending of the physical distances separating the components we distinguish two major cases:

INTRASYSTEM communication, which occurs within a single computer system and involves information transfer over the distance of not more than a meter or so.

INTERSYSTEM communication, which involves communication over long distance.

1.4.1 MODES OF COMMUNICATION:

There are two modes of communication parallel mode and serial mode. In short distance communication parallel mode is the general choice. In long distance, serial is the suitable mode.

In parallel mode, the interconnection cable is a bus with separate lines for al the bits of the bytes, lines for handshaking signals, lines for error checking signals such as parity. Thus the interconnection cable is expensive.

In serial mode, the interconnecting cable is a bus with two lines for receiving and transmitting the bit, lines for handshaking signals and line error checking signals.

Two types of serial transfer are in existence, asynchronous and synchronous transmission. The asynchronous format is character oriented.

Each character carries the information of the start and stop bits. This is also known as framing. This format is generally used in high-speed transmission.

Serial communication also can be classified according to the direction and simultaneity.

In simplex transmission data are transmitted in only one direction. In duplex transmission occurs in both directions. However if the transmission goes way at a time it is called half duplex if it goes in both ways simultaneously it is called gull duplex.

The rate are which the bits are transmitted/seconds is called baud in serial transmission.

1.4.2 COMPUTER PORTS:

This is an introduction to program the PC's communication ports. The basic way by which the computer communicates with the devices is through ports. On PC's it is basically an 8-bit doorway. The computer communicates with other devices either through serial ports of through parallel ports.

Each method has it's own merits and demerits. A parallel port transfers entire byte at a time whereas the serial port transfers only one bit at a time. This is the basic difference between the ports. These serial ports take eight-fold time that of a parallel post to transfer a byte. This makes us to come to a conclusion that parallel transfer is much faster than serial transfer is. In spite of this we have selected serial port as the medium of transfer for this project.

1.4.3 SERIAL AND PARALLEL COMMUNICATION:

Historically, the need for serial port arose when mainframe became large and fast enough to share time between several programs or tasks, rapidly switching between them to give each of them an illusion of exclusive attention. The multitask facilities each user to work on an independent terminal. These works even in case where independent terminal are at some distances from the main system and terminal as per requirements.

The main drawback of parallel communication, which forced people to use serial communication in case of long distances, is the high cost due to the need for separate wires for each bit to be transmitted simultaneously Another serious drawback in cases of long distances is the speeds at which the bits travel along the different wires, which differ slightly. Thus the bits reach the receiver at different times even together. Over the long distances these garbles the message.

Serial communication does not require use of expensive cables. The phone line transmits the data over long distances. To expand the communication over thousands of kilometers the usage of modems at both ends are the only additional requirements.

SYSTEM STUDY

2. SYSTEM STUDY

2.1 Interfacing

PC communicates with periperals through serial ports com1 or com2 which communicates the data in terms of pulse form as follows.

data		pulse
0	------->	+12v
1	------->	-12v

the above logic is considered as PC Logic.

Micro Controller communicates with periperals through serial pins rxd or txd which communicates the data in terms of pulse form as follows.

data		pulse
0	------->	0v
1	------->	5v

the above logic is considered as TTL Logic.

RS232 is the interface between PC and microcontroller which converts PC logic into TTL logic.

It receives the data from PC (+12v and -12v)and converts into TTL logic then this given to Micro controller.

2.2 RS-232 serial interfacing

A major problem with RS-232C is that it can only transmit data reliable about 50 ft [16.4 m] at its maximum rate of 20,000 Bd. If longer lines are used, the transmission rate has to be drastically reduced. This limitation is caused by the open signal lines with a single common ground that are used for rs-232c.

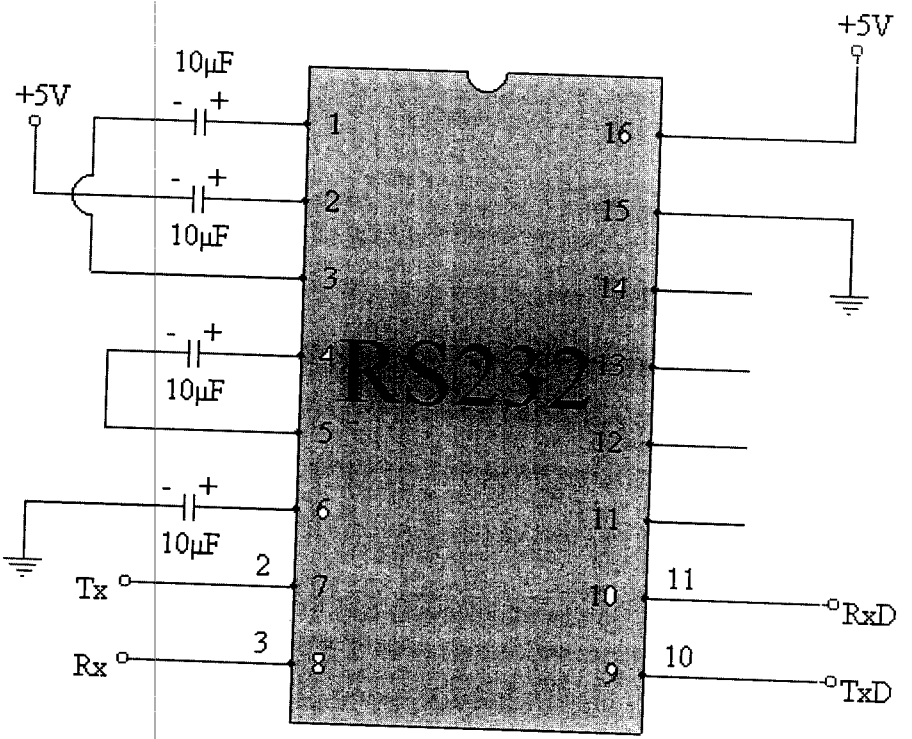
A newer standard, RS –422A specifies that each signal will be sent differentially over two adjacent wires in a ribbon cable or a twisted pair of wires as shown in Figure 13-11 a. Differential signals are produced by differential line drivers such as the MC 3487 and translated back to TTL levels by differential line receivers such as the MC3486. Data rates for this standard are 10 MBd for a distance of 50 ft [1220 m]. The higher transmission rate of RS-422A is possible because the differential lines are terminated by resistors so they act as transmission lines instead of simply open wires. A further advantage of differential signals is that any common-mode electrical noise induced in the two lines will be rejected by the differential line receiver. For RS-422A a logic high or mark is indicated by the B signal line being more positive than the A signal line. A logic low or space is indicated by the A signal line being more positive than the B signal line. The voltage difference between the two lines must be greater than 0.4 V, but not greater than 12 V. The common-mode voltage on the signal lines must be in the range of –7 V to +7 V.

Another EIA standard intended to solve the speed and distance problems of RS-232C is RS-423A. This standard specifies a low impedance signal-ended signal instead of the differential signal of RS-422A. The low-impedance signal can be sent over 50-Ω coaxial cable and terminated at the receiving end to prevent reflections. Figure 13-11b shows how an MC3487 driver and MC3486 receiver can be connected to produce the required signals. A logic high in this

represented by the A line being between 4 and 6 V negative with respect to the B line (ground), and a logic low is represented by the A line being 4-6 V positive with respect to ground.

The RS-422A and RS-423A standards do not specify connector pin numbers or handshake signals the way that RS-232C does. An additional EIA standard called RS-449 does this or the two. RS-449 specifies 37 signal pins on a main connector and 9 additional pins on an optional connector. The signals on these connectors are a superset of the RS-232C signals so adapters can be used to interface RS-232C equipment with RS-449 equipment. Still another EIA standard, RS-366, incorporates signals for automatic telephone dialing with modems.

2.3 RS-232 Signal Pins



89C51 MICROCONTROLLER

3. MICROCONTROLLER 89C51

3.1 Introduction:

Microcontrollers these days are silent workers in many apparatus, ranging from the washing machine to the video recorder. Nearly all of these controllers are mask programmed and therefore are of very little use for applications that require the programs to be changed during the course of execution.

Even if the programs could be altered, the information necessary to do so an instruction set, an assembler language and description for the basic hardware is either very difficult to obtain or are in adequate when it came to the issue of accessibility.

A marked exeception to the above category is the atmel 89C51 microcontroller belonging to the Atmel family. This microcontroller has features that seem to make it more accessible than any other single chip microcontroller with a reasonable price tag.

The 89C51, an 8 bit single chip microcontroller has got a powerful CPU optimized for control applications, 64K program memory address space, 64K data memory address space, 128 bytes of on chip RAM (read/write memory), for 8 bit bi-directional parallel ports one full duplex serial ports two 16 bit timers/counteres and an extensive interrupt structure.

The 98C51 is a second generation 8-bit single chip microcontroller. The 89C51 provides a significantly more powerful architecture, a more powerful instruction set and a full serial port.

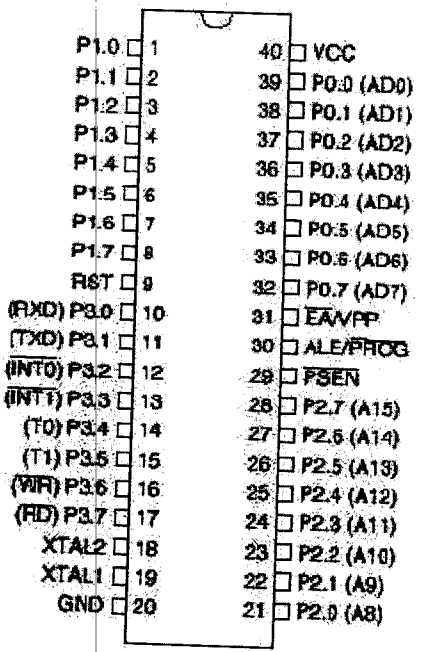
The 89C51 is a complete microcontroller. We will start our discussion on the programming model by looking at the architectural block diagram. As we can see the external connections very simple. The are 31 pins needed by the four 8 bit bi-directional ports. Eight additional pins are provide power, allow you to connect a crystal clock and provide a few timing and control signals.

The architecture includes the ALU, the accumulator, the stack pointer, a block of registers and a general purpose register-the B register. All these devices are connected to the 89C51 internal 8-bit data bus.

Each I/O port is also connected to the 8 bit internal data bus through a series of registers. These registers hold data during I/O transfers and control the I/O ports. The architectural block diagram also shows the 89C51 ROM and RAM.

Pin Configuration:

AT89C51



COMPARISON OF MICROPROCESSOR AND MICROCONTROLLER:

The difference between Microprocessor and Micro controller is Microprocessor can only process with the data, Micro controller can control external device. That is if you want switch "ON" or "OFF" a device, you need peripheral Ics to do this work with Micro controller you can directly control the device.

Like Microprocessor, Micro controller is available with different features. It is available with inbuilt memeory, I/O lines, timer and ADC. The micro controller, which we are going to use, is 89C51 it is manufactured by atmel, MC, USA. This is advanced version of 8031. This Micro controller have inbuilt 4K

bytes of flash ROM, 256 bytes of RAM, 32 I/O lines (4 bit ports) and 6 vectored interrupts.

3.2 Memory Organisation

FLASH ROM:

Flash ROM can be well explained with a block diagram as shown in the following figure. 4-kilo byte ROM is available in the Micro controller. It can be erased and reprogrammed. If the available memory is not enough for your program, you can interface the external ROM with this IC, it has 16 address lines, so maximum of (2^16) i.e. 64 bytes of ROM can be interfaced with this Micro controller. Both internal and external ROM cannot be used simultaneously.

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

RAM:

Internal 256 bytes of RAM are available for user. This 256 bytes of RAM can be used along with the external RAM. Externally you can connect 64-kilo bytes of RAM with micro controller. In internal RAM first 128 bytes of RAM is available for user and the remaining 128 bytes are used as special function

registers (SFR). These SFR's are used as control registers for timer, serial port etc.

INPUT/OUTPUT PORTS:

There are four I/O ports available in AT89C51. They are port 0, port 1, port 2, port 3. All these ports are eight bit ports. All these ports can be controlled as eight-bit port or it can be controlled individually. One of the main feature of this micro controller is it can control the port pins individually. For example to control a LED we need to use one I/O line in Microprocessor with 8255 we have to use an eight bit port. In 89C1 port 1 is available for users Port 3 is combined with interrupts. This can be used as interrupts (or) I/O ports, port 2 & port 0 is combined with address bus and data bus.

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

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

3.3 Power Modes of ATMEL 89C51 Microcontroller :

To exploit the power savings available in CMOS circuitry. Atmel's Flash microcontrollers have two software – invited reduced power modes.

IDLE MODE;

The CPU is turned off while the Ram and other on – chip peripherals continue operating. In this mode current draw is reduced to about 15 percent of the current drawn when the device is fully active.

POWER DOWN MODE:

All on – chip activities are suspended while the on – chip RAM continues to hold its data. In this mode, the device typically draws less than 15 μA and can be as low as 0.6μA.

POWER ON RESET:

When power is turned on, the circuit holds the RST pin high for an amount of time that depends on the capacitor value and the rate at which it charges. To ensure a valid reset, the RST pin must be held high long enough to allow the oscillator to start up plus two machine cycles.

PROGRAM MEMORY:

As shown the map of the lower part of the program memory, after reset, the CPU begins execution from location 0000h.

As shown each interrupt is assigned a fixed location in program memory. The interrupt causes the CPU to jump to that location, where it executes the service routine. External Interrupt 0 for examples assigned to location 0003h. If the Interrupt is not used, its service location is available as general – purpose program memory.

The interrupt service locations are spaced at 8 byte intervals 0003h for external interrupt 0, 000Bh for Timer 0, 0013h for External interrupt 1, 001Bh fir Timer 1 and so on. If an Interrupt service routine is short enough (as is often the case in control applications) it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations. If other interrupts are in use.

The lowest address of program memory can be either in the on – chip Flash or in an external memory. To make this selection, strap the External Access (EA) pin to either Vcc or GND.

For example, in the AT89C51 with 4K bytes of on – chip Flash, if the EA pin is strapped to Vcc, program fetches to addresses 0000h through 0FFFh are directed to internal Flash. Program fetches to addresses 1000h through FFFFh are directed to external memory.

DATA MEMORY:

The Internal Data memory is divided into three blocks namely,

- The Lower 128 Bytes of Internal RAM.
- The upper 128 Bytes of Internal RAM.
- Special Function Registers.

Internal Data memory Addresses are always 1 byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in fact accommodate 384 bytes. Direct addresses higher than 7Fh access one memory space and indirect addresses higher than 7Fh access a different memory space.

The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word (PSW) Select, which register bank, is in use. This architecture allows more efficient use of code space, since register instructions are shorter than instructions are shorter than instructions that use direct addressing.

The next 16 – bytes above the register banks form a block of bit addressable memory space. The microcontroller instruction set includes a wide selection of single – bit instructions and this instruction can directly address the 128 bytes in this area. These bit addresses are 00h through 7Fh.

PROGRAM STATUS WORD:

Program Status Word Register in Atmel Flash Microcontroller

CY	AC	FO	RS1	RS0	OV	----	P
PSW7							PSW0

PSW0: Parity of Accumulator Set By Hardware to 1 if it contains an Odd number of 1s otherwise it is reset 0.

PSW1: User Definable Flag.

PSW2: Overflow Flag Set By Arithmetic Operations.

PSW3: Register Bank Select.

PSW4: Register Bank Select.

PSW5: General Purpose Flag.

PSW6: Auxiliary Carry Flag Receives Carry Out from Bit of Addition Operands.

PSW7: Carry Flag Receives Carry Out From Bit 1 of ALU Operands.

3.4 Interrupts:

The AT89C51 Provides 5 interrupts sources: Two External interrupts, two-timer interrupts and a serial port interrupt.

The External Interrupts INT0 and INT1 can each either level activated or transition – activated, depending on bits IT0 and IT1 in register TCON. The Flags that actually generate these interrupts are the IE0 and IE1 bits in TCON. When the service routine is vectored to hardware clears the flag that generated an external interrupt. If the interrupt was transition – activated. If the interrupt was level – activated, then the external requesting source (rather than the on – chip hardware) controls the requested flag.

The Timer 0 and Timer 1 interrupts are generated by TF1 and TF0 which are set by a roll over in their respective Timer / counter Register (except for Timer 0 in Mode 3). When a timer interrupt is generated it when the service routine is vectored to.

The logical OR of RI and TI generate the Serial Port Interrupt. Neither of these flag is cleared by hardware when the service routine is vectored to. In fact, the service routine normally must determine whether RI or TI generated the interrupt and the bit must be cleared in software.

IE: INTERRUPT ENABLE REGISTER:

EA	---	ET2	ES	ET1	EX1	ET0	EX0

Enable bit = 1 enabled the interrupt

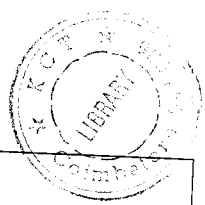
SYMBOL	POSITION	FUNCTION
EA	IE.7	Global enable / disable all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enabled bit.
---	IE.6	Undefined / reserved.
ET2	IE.5	Timer 2 Interrupt enables bit.
ES	IE.4	Serial Port Interrupt enabled bit
ET1	IE.3	Timer 1 Interrupt enables bit.
EX1	IE.2	External Interrupt 1 enables bit.
ET0	IE.1	Timer 0 Interrupt enables bit.
EX0	IE.0	External Interrupt 0 enables bit.

IE: INTERRUPT PRIORITY REGISTER:

---	---	PT2	PS	PT1	PX1	PT0	PX0

Enable bit = 1 enabled the interrupt.

Enable bit = 0 disables it.



SYMBOL	POSITION	FUNCTION
---	IP.7	Reserved.
---	IP.6	Reserved.
PT2	IP.5	Timer 2 Interrupt priority bit.
PS	IP.4	Serial Port Interrupt priority bit
PT1	IP.3	Timer 1 Interrupt priority bit.
PX1	IP.2	External Interrupt 1 priority bit.
PT0	IP.1	Timer 0 Interrupt priority bit.
PX0	IP.0	External Interrupt 0 priority bit.

PRIORITY LEVEL STRUCTURE:

Each interrupt source can also be individually programmed to one of two priority levels by setting or clearing a bit in special function register IP (interrupt priority) at address OB811. IP is cleared after a system reset to place all interrupts at the lower priority levelly default. A low – priority interrupt can be interrupted by a high – priority interrupt but not by another low – priority interrupt. A high – priority interrupt can be interrupted by any other interrupt sources.

If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If request of the same priority lev el are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is second priority structure determined by the polling sequence as follows.

OSCILLATOR AND CLOCK CIRCUIT:

XTAL1 and XTAL2 are the input and output respectively of an inverting amplifier, which is intended for use as a crystal oscillator in the pierce configuration. In the frequency range of 1.2 MHz to 12 MHz. XTAL2 is also the input to the internal clock generator.

To drive the chip with an internal oscillator, one would ground XTAL1 and XTAL2. Since the input to the clock generator is a divide – by two flip – flops, there is no requirement on the duty cycle of the external oscillator signal. However, minimum high and low times must be observed.

The clock generator divides the oscillator frequency by 2 and provides a two-phase clock signal to the chip. The phase 1 signal is active during the first half jof each clock period and the phase 2 signals is active during the second half of each clock period.

CPU TIMING:

All ATMs Flash microcontroller have an on – chip oscillator, which can be used, as the clock source for the CPU. To use the on – chip oscillator, connect a crystal or ceramic resonator between the XTAL1 and XTAL2 pins of the microcontroller and connect the capacitors to ground as shown.

The internal clock generator defines the sequence of states that make up the microcontroller machine cycles.

3.5 Timers:

Two special function registers, TMOD and TCON, are used to define operating modes and control the functions of the timer / counters. When an instruction changes the content of TMOD or TCN, the change is latched into the special function register and takes effect at SIPI of the nest instruction's first cycle. The register are shown below

TMOD: TIMER MODE CONTROL REGISTER

GATE	C/T	M1	M0	GATE	C/T	M1	M0

When M1, M0 specify the mode as follows

M1	M2	MODE	DESCRIPTION
0	0	0	13 Bit Counter
0	1	1	16 Bit Counter
1	0	2	8 Bit Counter with Auto Reload
1	1	3	Split Timer 0 into 8 bit Counters or Timer

SERIAL PORT CONTROL (SCON) SPECIAL FUNCTION REGISTER:

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

When SM0 and SM1 specify the mode as follows

SM1	SM0	MODE	OPERATION
0	0	0	Shift register; baud = f/12
0	1	1	8 – bit UART; baud = variable
1	0	2	9 – bit UART; baud = f/32 or f/64
1	1	3	9 – bit UART; baud = variable

SM2: Multiprocessor communications bit Set / cleared by program to enable multiprocessor communications in modes 2 and 3.

REN: Receive enable bit. Set to 1 to enable reception; cleared to 0 to disable reception.

TB8: Transmitted bit 8. Set / cleared by program in modes 2 and 3.

RB8: Received bit 8. Bit 8 of received data in modes 2 and 3; stop bit in mode 1.

TI: Transmit interrupt flag. Set to one at the end of bit 7 time in mode 0 and at the

beginning of the stop bit for the other modes.

RI: Receive interrupt flag. Set to one at the end off bit 7 time in mode 0 and halfway

through the stop bit for other modes.

THE POWER MODE CONTROL (PCON) SPECIAL FUNCTION REGISTER:

SMOD	---	---	---	GF1	GF0	PD	IDL

SYMBOL	FUNCTION
SMOD	Serial baud rate modify bit. Set to 1 by program to double baud rate using timer 1 for modes 1, 2 and 3. Clear to 0 by program to use timer 1 baud rate.
-----	Not implemented.
GF1	General purpose user flag bit 1. Set / Cleared by program
GF0	General purpose user flag bit 0. Set / Cleared by program
PD	Power down bit. Set to 1 by program
IDL	Idle mode bit. Set to 1 by program.

3.6 Addressing Modes:

The 89C51 instruction operate on data stored in internal CPU registers, external memory or an I/O ports. There are number of methods (modes) in which these registers, memory (internal or external0 and I/O ports (internal or external) can be addressed called addressing modes. These modes are

- Immediate
- Direct
- Indirect
- Register
- Register Specific
- Indexed

IMMEDIATE ADDRESSING:

In this mode, the data to be operated upon is in the location immediately following the opcode. For example the instruction

MOV A,#41

DIRECT ADDRESSING:

In direct addressing, the operand is specified by an 8-bit address field in the instruction. Only internal RAM and SFR's can be directly addressed. For example the instruction,

INC 20

Increments the contents of the chip data address by one.

INDIRECT ADDRESSING:

In indirect addressing, the instruction specifies a register, which contains the address of the operand. Both external and internal RAM can be indirectly addressed.

The address register of 8 bit addresses can be R1 or R1 of the selected register bank of the stack pointer. The address register for 16-bit addresses can only is a 16-bit data pointer register, DPTR. For example the instruc

MOVX A, @DPTR

Writes the contents of the accumulator to the address held by the DPTR register.

REGISTER ADDRESSING:

The register banks, containing registers R0 to R7, can be addressed by certain instructions which carry a three bit register specific within the opcode of the instruction.

Instructions that access the registers this way are code efficient, since this mode eliminates an address byte.

When the instruction is executed, one of the eight registers in the selected bank is accessed. One of the four banks is selected at the execution time be the two bank select bits in the PSW. For example the instruction

MOV A,R0

Copies the contents of the register R0 (of the selected bank) to the accumulator.

REGISTER SPECIFIC INSTRUCTION:

Some of the instructions are specific to a certain register. Stating in other words, the opcode itself contains the implied operand or the operand destination. For the example some instruction always operates on the accumulator or the data pointer etc., so no address byte is needed to point to it. The opcode itself does that instructions that refer to the accumulator as A assemble as accumulator – specific opcodes. Such instructions are byte long. These instructions are operated on the accumulator.

RRA

This instruction, when executed rotates the contents of the accumulator one bit towards the right.

INDEXED ADDRESSING:

Only the program memory can accessed with the indexed addressing and it can only be read. This addressing mode is indented for reading lookup tables in the program memory. A 16 bit base register (either DPTR or the program counter) points to the base of the table and the accumulator is setup with the table entry number in the program memory is formed by adding the accumulator data to the

base pointer. The instruction reads the contents of the program memory, whose address is obtained by adding the contents of the DPTR and accumulator.

MOVC A,@A+DPTR

Another type of indexed addressing is used in the 'case jump' instruction. In this case the destination address of the jump instruction is computed as the sum of the contents of the contents of the base pointer and the accumulator.

PATIENT CARE SYSTEM

4. PATIENT CARE SYSTEM

4.1 Temperature Sensing Unit

We are using Thermister as a sensor to measure temperature.The principe of thermistor is , If the temperature is increses ,the internal resistance of the sensor will decrese vice versa. The sensor is connected with wheatstone bridge.Initially we should keep the bridge in balanced condition ,we tune the resistors for balance condition, the resistance value of the wheatstone bridge get some change due to the temperature change in the thermister , in this moment the bridge get unbalanced and we get unbalanced differential output. The output of the bridge is given to differential amplifier, the output of the bridge is in the form of millivolts and we get the differential voltage by using the differential amplifier.

The formula for the wheatstone bridge is in balance condition.

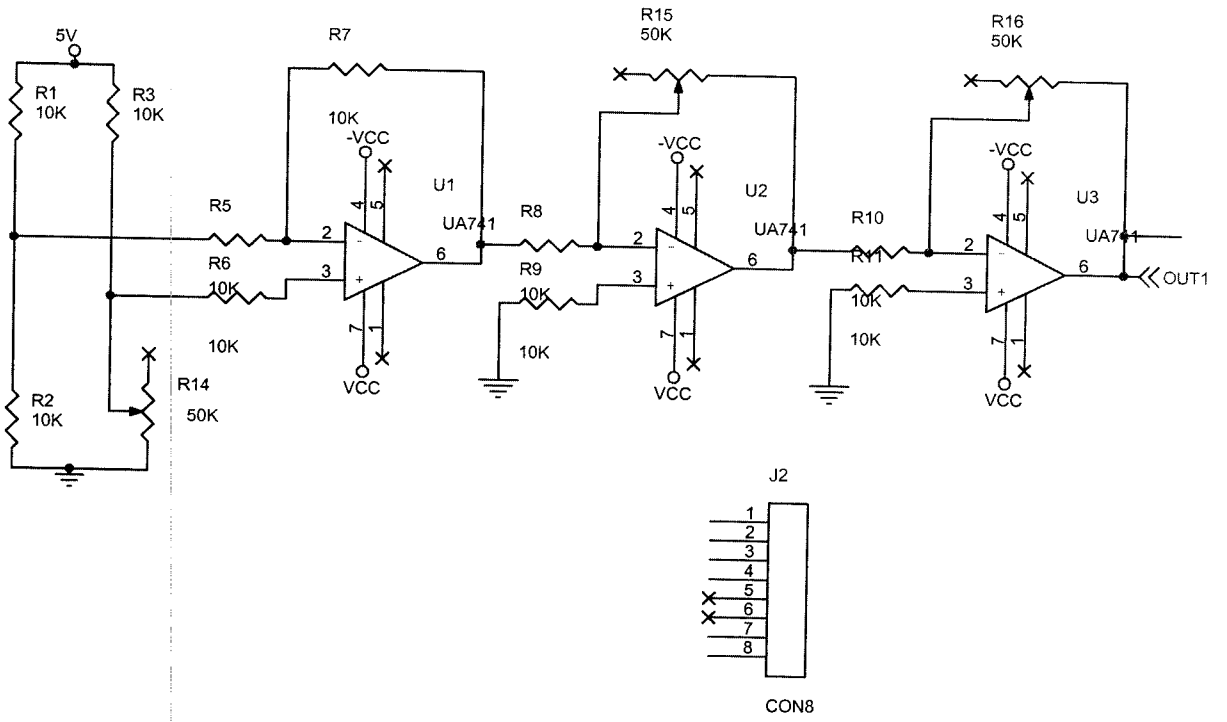
$$R1R3=R2R4$$

Where,

R1,R2,R3 = fixed resistance

R4 = The internal resistance of the sensor

The output of the differential amplifier is in the form of millivolts, so we have to amplify the millivolts into volts by using the inverting amplifier stages. The amplified voltage is given to the A/DConverter, it converts the analog signal into the corresponding digital value and it gives the digital output as input to the the micro controller.



4.2Drips Monitoring

Infra Red sensor

Infra red Sensor → TTL logic converter --> Driver Circuit

When the supply gives to the IR transmitter ,it will emits the IR rays which is received by IR receiver.The output from the IR receiver is connected to the Amplifier and drives by driver which is converted into TTL logic by TTL logic converter.

Principle of Infra Red Sensor:

When receiver receives the signal , the internal resistance of the IR sensor decerses and vise versa.So the voltage across the Infra red sensor varies

Principle of Amplifier:

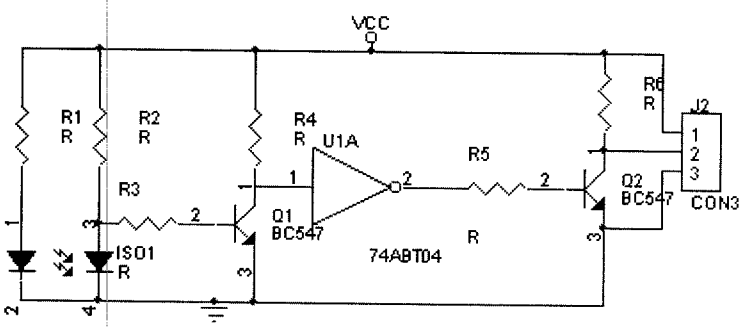
Amplifier is a circuit which amplifies the input signal.Transister acts as a amplifier which gets the input from the Infra red sensor.

Principle of Driver:

Driver is a electronic component which drives the signal without any loss in the input signal. Logic gates acts as a driver which gets the input from the amplifier.

Principle of TTL logic converter:

TTL logic converter is a converter which converts input analog signal into digital signal (0's & 1's).It gets the input from the driver.



4.3 Respiration Measurement

BASIC INFORMATION OF OP-AMP:

The op – amp consist of two input terminals and one output terminal. The terminal with a (-) sign is called inverting input terminal and the terminal with (+) sign is called the non-inverting input terminal.

INVERTING AMPLIFIER:

The inverting amplifier circuit is shown. The output voltage V_o is fed back

feedback resistor. Input signal V_i is applied to the inverting input terminal through R_i and non-inverting input terminal of op-amp is grounded.

For simplicity, assume an ideal op-amp. As $V_d = 0$, mode 'a' is at ground potential and the current I_1 through R_1 is

$$I_1 = \frac{V_i}{R_1}$$

Also since op-amp drawsno current flowing through R_i must flow through R_f. The output voltage,

$$V_o = - I_i R_f = -V_i = \frac{R_f}{R_i}$$

Hence, the gain of the inverting amplifier (also referred as closed loop gain) is,

$$A_{CL} = \frac{V_o}{V_i} \quad \frac{R_f}{R_1}$$

Alternatively, the nodal equation at the node 'a' is

$$\frac{(V_a - V_i)}{R_1} + \frac{(V_a - V_o)}{R_f} = 0$$

Where V_a is the voltage at node 'a'. Since node 'a' is at virtual ground, $V_a = 0$ therefore, we get

$$A_{CL} = \frac{V_o}{V_i} = - \frac{R_f}{R_1}$$

The negative sign indicates a phase shift of 180^{o} between V_i and V_o. Aslo since inverting input terminal is at virtual ground, the effective input impedance is R_1.

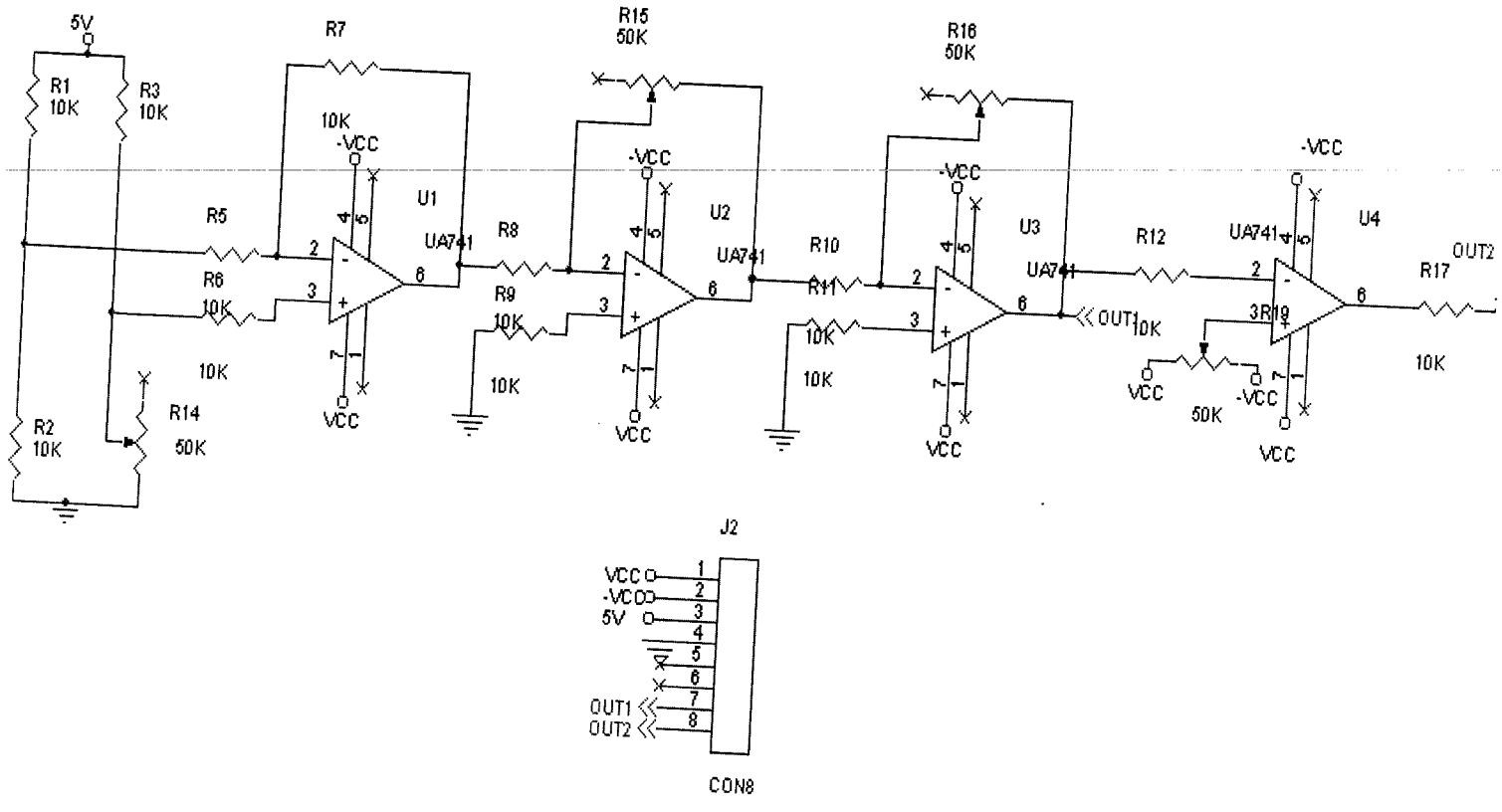
The value of R_1 should be kept fairly large to avoid loading effect. This however, limits the gain that can be obtained from this circuit. A load resistor R_l is usually put at the output in actual practice.

COMPARATOR:

A comparator is a circuit which compares a signal voltage applied at one input of an op-amp with a known reference voltage at the other input. It is basically an open loop op-amp with output $\pm V_{sat}$ as shown in the ideal transfer characteristics from fig. However, acommercial op-amp has the transfer characteristics from the fig. It may be seen that the change in the output state takes place with an increment in input V_i of only 2mV. This is the uncertainly region where output cannot be directly defined. There are basically two types of comparators:

The circuit from fig is called a non-inverting comparator. A fixed reference voltage V_{ref} is applied to (-) input and a time varying signal V_i is applied as (+) input terminal as shown. Thus a V_{ref} of desired amplitude and polarity can be obtained by simply adjusting the 10 KΩ potentiometer.

A pratical inverting comparator in which the reference voltage V_{ref} is applied to the (+) is applied to the (+) input and V_i is applied tok (-) input. For a sinusoidal input signal, the output waveform is shown fig. V_{ref} positive and negative respectively.


5V
R1
10K
R3
10K
R2
10K
R14
50K
R5
10K
R6
10K
R7
10K
-VCC
U1
UA741
VCC
R8
10K
R9
10K
R15
50K
-VCC
U2
UA741
VCC
R10
10K
R11
10K
R16
50K
-VCC
U3
UA741
VCC
OUT1
R12
10K
-VCC
U4
UA741
3R19
50K
VCC
-VCC
VCC
R17
10K
OUT2
J2
VCC
-VCC
5V
OUT1
OUT2
1
2
3
4
5
6
7
8
CON8

HARDWARE

5. HARDWARE

5.1 Analog To Digital Convertor ADC(0809)

Every sensor output is in the form of analog voltage.Microcontroller receives only in the form of digital values.So we have to interface the sensor with the microcontroller we need the A/D convertor.

In this ADC has eight analog channel inputs,eight digital outputs,three channel selection lines and SOC (Start Of Conversion),EOC(End Of Conversion) signal lines.

i) First we have chose which channel to be used as follows:

selection lines			channel
A2	A1	A0	selection
0	0	0	ch0
0	0	1	ch1
0	1	0	ch2
0	1	1	ch3
1	0	0	ch4
1	0	1	ch5
1	1	0	ch6
1	1	1	ch7

ii) we have send the SOC signal from the microcontroller to the ADC

iii) we have receive the EOC signal from the ADC to microcontroller

iv) we have to read the Digital values from the ADC corresponding analog value.

v) The range of input voltage of ADC in the range of 0v---5v .

vi) The range of output from the ADC in the range of 00h---ffh .

5.2 DTMF Generator

We are sending the datas from The microcontroller to the receiver side by the FM communication.In this communication we are using the DTMF technique(Dual Tone Multi Frequency).

The generator UM91214b generates the frequency for the corresponding datas.Micro controller sends the data through the DTMF generator to the FM transmitter.FM transmitter receives the frequency from the DTMF generator and send thisFrequency to the antenna.

5.3 DTMF Decoder

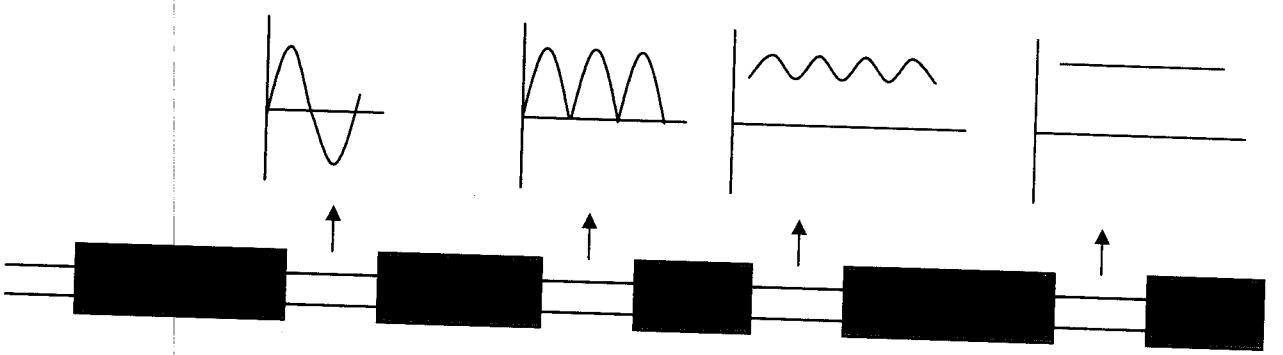
Micro controller sends the data through the DTMF generator to the FM transmitter.FM transmitter receives the frequency from the DTMF generator and send thisFrequency to the antenna.Receiving antenna receives the corresponding frequency and given To the DTMF decoder. The DTMF(8870) decoder decodes the binary data for the corresponding frequency.

5.4 Power Supply Description

The power supply circuit for 5V and +/- 12V supplies is shown in figures.

filters, rectifiers and then voltage regulators. Starting with an ac voltage, a steady dc voltage is obtained from an IC voltage regulator unit, which takes a dc voltage and provides a somewhat lower dc voltage, which remains the some even if the input dc voltage varies or the output load connected to the dc voltage changes.

A block diagram containing the parts of a typical power supply and the voltage at various points in the unit is shown. The ac voltage, typically 120V rms, is connected to a transformer, which steps that ac voltage down to the level for the desired dc output. A diode rectifier then provides a full – wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. The resulting dc voltage usually has some ripple voltage but also remains the same dc value even if the input dc voltage varies somewhat, or the load connected to the output dc voltage changes. This voltage regulation is usually obtained using one of a number popular voltage regulator IC units.



IC VOLTAGE REGULATORS:

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

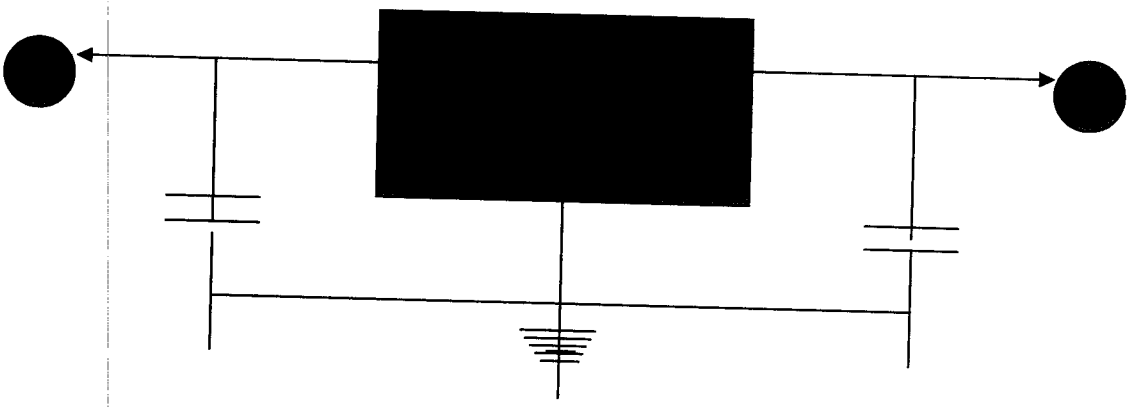
A power supply can be built using a transformer connected to the ac supply line to step the ac voltage to a desired amplitude

using an IC regulator. The regulators can be selected for operation with load currents from hundreds of milli amperes to tens of amperes, corresponding to power ratings from mill watts to tens fo watts.

THREE – TERMINAL VOLTAGE REGULATORS:

Figure shows the basic connection of a three – terminal voltage regulator IC to a load. The fixed voltage regulator has an unregulate4d dc input voltage, Vi, applied to one input terminal, a regulated output dc voltage, Vo, from a second terminal, with the third terminal connected to ground. For a selected regulator, IC device specifications list a voltage range over which the input voltage can vary to maintain a regulated output voltage over a range of load current. The specifications also list the amount of output voltage change resulting from ka change in load current (load regulation) or in input voltage (line regulation).

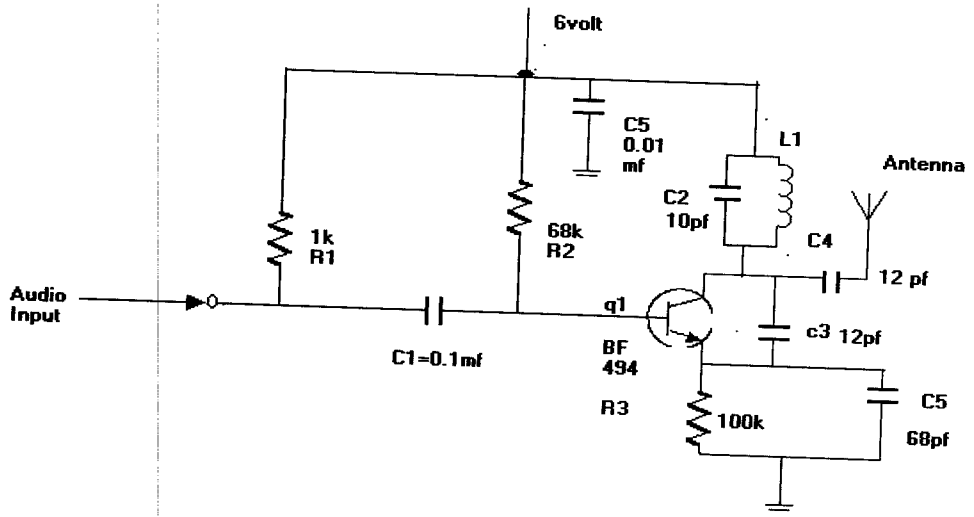
FIXED POSITIVE VOLTAGE REGULATORS:



The series 78 regulators provide fixed regulated voltages from 5 to 24V. Figure shows how one such IC, a 7812, is connected to provide voltage regulation with output from this unit of +12V dc. An unregulated input voltage Vi is filtered by capacitor C1 and connected to the IC's IN terminal. The IC's OUT terminal provides a regulated +12V, which is filtered by capacitor C2 (mostly for any high – frequency noise). The third IC terminal is connected to ground (GND). While the input voltage may vary come permissible voltage range and the output load

may vary over some acceptable range, the output voltage remains constant within specified voltage variation limits.

5.5FM TRANSMITTER



A simple but practical FM transmitter circuit is given in the figure. This circuit can be operated 9v supply. The signals are passed from the encoder circuit to the base of the transistor Q1 through the capacitor c1.Q1 is a NPN transistor and is collector is given positive supply through the resistance R3 while its base is given the forward supply through the resistance R2 ,the voltage signals are given to base of the transistor Q1 through the capacitor c1.

A capacitor c3 has also been connected between the collector and the emitter of transistor q1.This capacitor triggers the oscillations . As soon as the Audio signal is received at the base then the transistor starts to oscillate and is

5.6 FM RECIEVER

IC CXA1019s is a thirty pin DIL IC.Facility to connect a tuning indicator has been provided in this IC apart from the various sections built within the IC.A 10.7 Mhz ceramic filter has been used in this circuit in place of the IFTS.An audio output section is also built within the IC apart from all the necessary sections.The audio output section of this IC has not been used in the give circuit.The signal received at the IC pin number 24 is given to the pin 1 of the volume control and the pin2 of the volume controllers connected to the audio output sections.This section is not shown in the given circuit.

The IC pin no:27 is the positive supply pin which is given 9 volt supply through a resistance R4.This supply is filtered by the capacitor c1 .IC pin no:7 and * are connected to the oscillator section and a coil L1 is connected at the pin no:8.The capacitor c3 and a button trimmer B1 are connected parallel to this coil L1.The desired frequency is selected by this trimmer RF coil L2 is connected at the IC pin no 10 and a capacitor c7 and a button trimmer B2 are connected in parallel to this coil.

SOFTWARE

6. SOFTWARE

6.1 Lab View

Laboratory Virtual Instrument Engineering Workbench (Lab VIEW) developed by national instruments, is a powerful analysis programming language. It was first introduced in 1983, when national instrument was interested in discovering new ways to decrease the time used in programming software language. The idea was to combine front panel user interface with block diagram programming methods allowing for fast and more efficient, software based graphical instrumentation. Since 1983 Labview has grown popular in many research laboratories and industries. Labview is a part of a new graphical programming language called 'G'. It uses terminology, icons and ideas that are familiar to technician, scientist and engineers. Virtual instrument (VIs) created in labview can generate charts, graphs and customized, user defined graphics. Labview integrates data acquisition, analysis and presentation in one system making programming simple and manageable.

INTRODUCTION TO LAB VIEW 6.1

Lab view is a graphical programming language (aimed at scientists and engineers)
that uses icons instead of line of next to create applications. In contrast to text-based programming languages, where instructions determine program execution, Labview uses dataflow programming, where the flow of data determines execution.

In Lab view, you build a user interface by using a set of tools and objects. The user interface is known as the front panel. You then add code using graphical representations of functions to control the front panel objects. The block diagram contains this code. In some ways, the block diagram resembles a flowchart.

It has built-in functionality for data acquisition, image acquisition, motion

and data presentation. Lab view gives you the flexibility of powerful programming languages without the complexity of traditional development environments and delivers extensive ACQUISITION, ANALYSIS AND PRESENTATION capabilities within a single environment, so you can seamlessly develop a complete solution on the platform of your choice.

INTRODUCTION TO VIRTUAL INSTRUMENTS:

Lab view programs are called virtual instruments, or VIs, because they imitate real instruments only that they are virtual instruments. VIs has three parts: the front panel, block diagram and the icon/connector.

Front panel and block diagram have identical menu bar and tools icons except that for front panel there is a controls menu whilst for block diagram there is a Functions menu. Controls menu contributes to controls and indicators; they represent the input parameters and output values respectively. Functions menu contributes to functions. You can also the controls or functions menu by clicking the right button of the mouse anywhere inside the windows.

FRONT PANEL -Serves as the user interface.

The front panel is the user interface of the VI.You builds the front panel with controls and indicators, which are the interactive input and output terminals of the VI, respectively.Controls are knobs, push buttons, dials and other input devices. Indicators are graphs, LEDs, and other displays. Control simulates instrument input devices and supply data to the block diagram of the VI.
Indicators simulate instruments output devices and display data the block diagram acquires or generates.

6.2 Coding

```
;-------------------------------------
        ; Drips P3.2 , res -p3.3
        ; ADC (SOC) - p3.6
        ;-------------------------------------

        org   0000h
        ljmp  main

        org   0003h
        ljmp  inter0

        org   000bh
        ljmp  timer


        org   0013h
        ljmp  inter1

        org   001bh
        reti

        org   0023h
        reti


        org    100h
main:   mov    p0,#00h
        mov    p2,#00h
        mov    r3,#00h      ;RES
        mov    r4,#00h      ;DRIPS
        mov    r5,#00h      ;TEMP
        lcall  timer_init
        sjmp   $


timer:  clr    tr0
        inc    r0
```

```
        cjne  r0,#10h,jjump
        mov   r0,#00h

        lcall htod_res
        lcall trans_res
        lcall delay
        lcall delay
        lcall delay
        lcall delay
        lcall htod_drip
        lcall trans_drip
        lcall delay
        lcall delay
        lcall delay
        lcall delay
        lcall temp_in
        lcall htod_temp
        lcall trans_temp
        lcall delay
        lcall delay
        lcall delay
        lcall delay

        mov    r3,#00h      ;RES
        mov    r4,#00h      ;DRIPS
        mov    r5,#00h      ;TEMP

jjump:  mov   th0,#0bh
        mov   tl0,#0dbh
        setb  tr0
        reti




temp_in:
      setb  p3.2
      nop
      nop
```

```
        nop
        nop
        nop
        clr  p3.2

        lcall delay

        mov  a,p1
        mov  r5,a
        ret



timer_init:
           mov  tmod,#21h
           mov  th1,#0e8h
           mov  th0,#0bh
           mov  tl0,#0dbh
           mov  scon,#58h
           setb ea
           setb es
           setb ex0
           setb ex1
           setb it0
           setb it1
           setb et0
           setb et1
           setb tr0
           setb tr1
           ret


 inter0:   inc  r2  ;drips
           reti


 inter1:   inc  r3  ;RES
           reti
```

```
htod_res:
          mov  a,r3    ;RES
          mov  b,#64h
          div  ab

          mov  63h,a
          orl  a,#20h
          mov  p0,a

          mov  a,b
            mov  b,#0ah
          div  ab

          mov  64h,a
          mov  65h,b
          swap a
          orl  a,b
          mov  p2,a
          ret


trans_res:
          mov   a,#'B'
          mov   sbuf,a
          mov   scon,#58h
          lcall delay

          mov   a,63h
          add   a,#30h
          mov   sbuf,a
          mov   scon,#58h
          lcall delay

            mov   a,64h
          add   a,#30h
          mov   sbuf,a
          mov   scon,#58h
          lcall delay
```

```
          mov   a,65h
         add   a,#30h
         mov   sbuf,a
         mov   scon,#58h
         lcall delay
          ret

htod_drip:
          mov  a,r4   ;DRIPS
          mov  b,#64h
          div  ab
          mov  66h,a
          orl  a,#30h
          mov  p0,a
          mov  a,b
            mov  b,#0ah
          div  ab
          mov  67h,a
          mov  68h,b

          swap a
          orl  a,b
          mov  p2,a
          ret


trans_drip:
         mov   a,#'C'
         mov   sbuf,a
         mov   scon,#58h
         lcall delay
         mov   a,66h
         add   a,#30h
         mov   sbuf,a
         mov   scon,#58h
         lcall delay
           mov   a,67h
         add   a,#30h
         mov   sbuf,a
```

```
        lcall delay
          mov   a,68h
        add   a,#30h
        mov   sbuf,a
        mov   scon,#58h
        lcall delay
          ret


htod_temp:
          mov  a,r5    ;TEMP
          mov  b,#64h
          div  ab
          mov  69h,a
          orl  a,#40h
          mov  p0,a
          mov  a,b
            mov  b,#0ah
          div  ab
          mov  6ah,a
          mov  6bh,b
          swap a
          orl  a,b
          mov  p2,a
          ret

trans_temp:
         mov   a,#'D'
         mov   sbuf,a
         mov   scon,#58h
         lcall delay
         mov   a,69h
         add   a,#30h
         mov   sbuf,a
         mov   scon,#58h
         lcall delay
           mov   a,6ah
        add   a,#30h
        mov   sbuf,a
```

```
        mov   scon,#58h
        lcall delay
          mov   a,6bh
        add   a,#30h
        mov   sbuf,a
        mov   scon,#58h
        lcall delay
          ret


delay: mov   70h,#0ffh
del:   mov   71h,#0ffh
dell:  djnz  71h,dell
       djnz  70h,del
       ret
```

FUTURE ENHANCEMENT

7. FUTURE ENHANCEMENT

The project can be enhanced to transmit the parameters itself that are recorded, in wireless to the doctor who can access the information through a device customized in this application. Also the project can be used in industrial application to keep monitor of time sensitive parameters that are to be taken care immediately based on the variation like temperature monitoring in a power plant system.

CONCLUSION

8. CONCLUSION

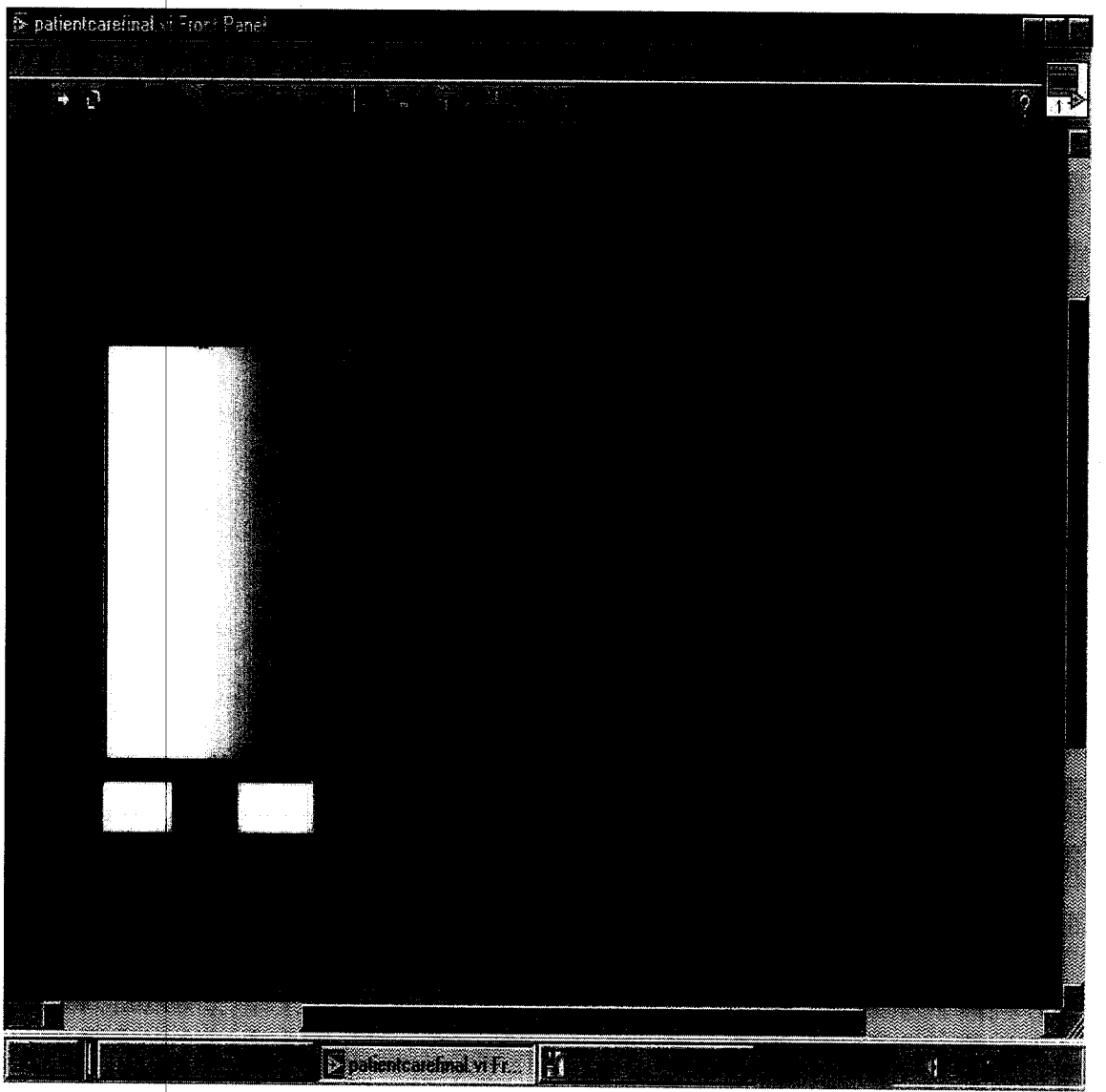
The complete system is more reliable than the existing manual methods in the industrial process. The integration of hardware and software provide efficient operation of the system. Moreover the developed system is quite simple and user friendly. The system has been working satisfactorily with the designed values and the system will fulfill the industrial requirement.

APPENDIX

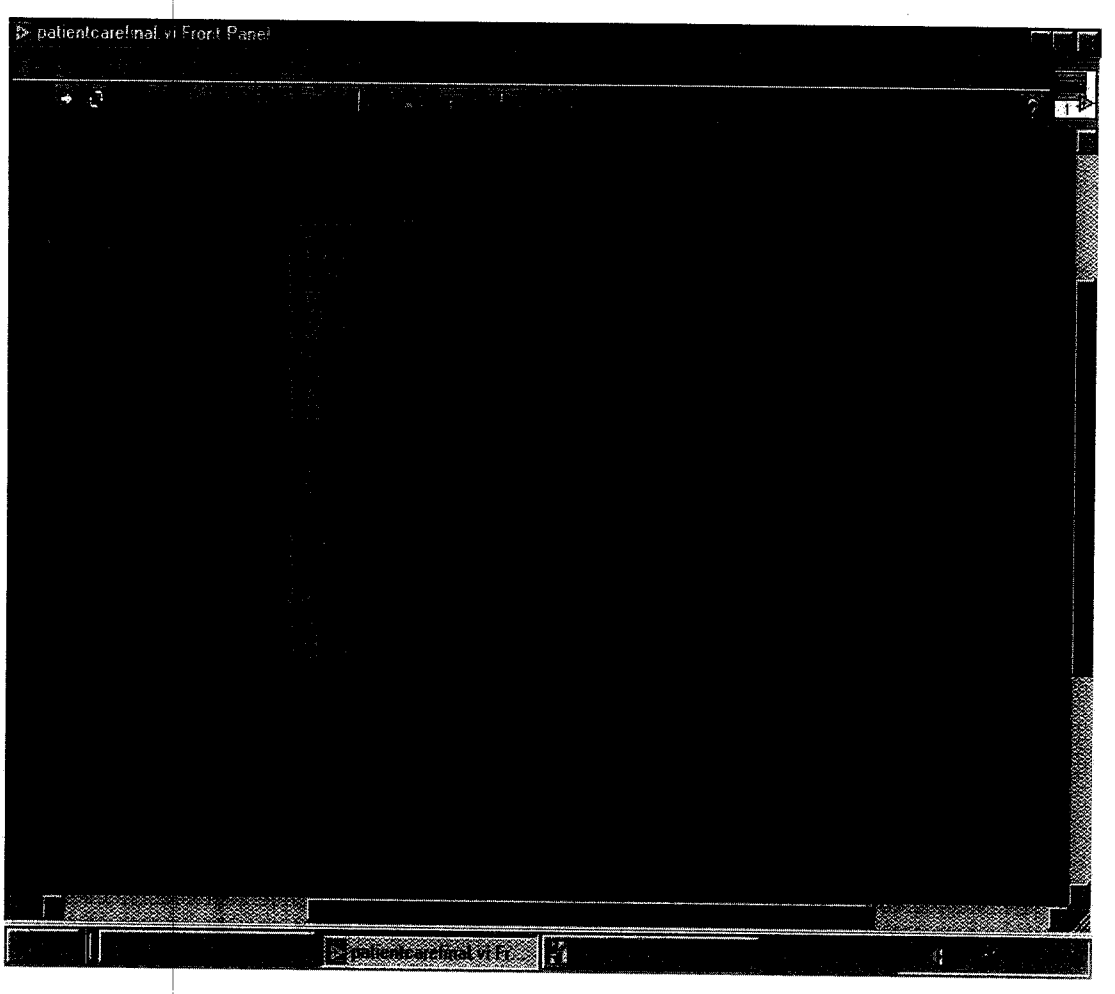
9. APPENDIX

Body Temperature

Drips Monitoring



piration



Analog To Digital Conversion :

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.)

10. BIBILOGRAPHY

- PeatMan,Design With Microcontroller-LPE Publish2000
- Kenneth J. Ayala, The 8051 Microcontroller-PRI Publishers
- Simon Haykin, Communication System-John Wiley & Sons Publishers
- www.Discovercircuits.com
- www.Electronicscircuits.com