**Applications of 8051 Microcontrollers**

**INTRODUCTION:**

A microcontroller is a versatile chip which can be used in various fields starting from simple consumer electronics to high end medical, automobile and defense applications also. So, now a days the microcontrollers are found in every walk of life. In the lab you can use a microcontroller to measure accurately the voltage, current and resistances also . The various fields are listed below.

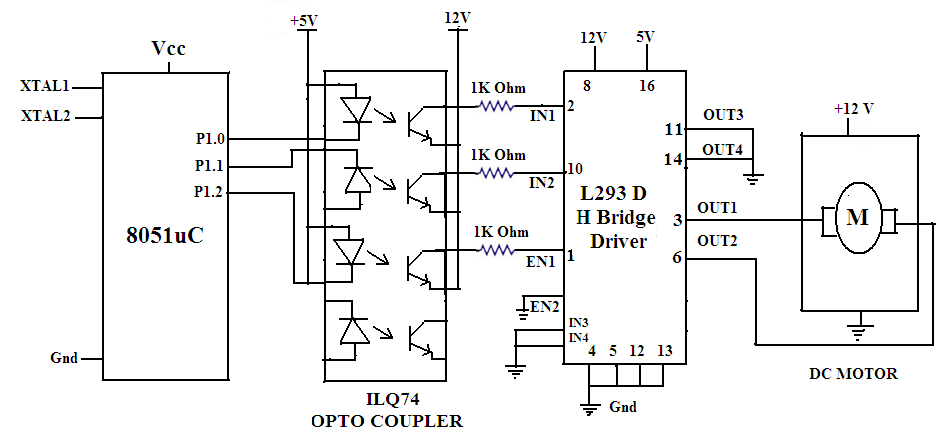
* Automobile
* Aeronautics
* Space
* Rail Transport
* Mobile communications
* Industrial processing
* Remote sensing ,Radio and Networking
* Robotics
* Consumer electronics , music players, Computer applications
* Security (e-commerce, smart cards)
* Medical electronics (hospital equipment, and mobile monitoring) and
* Defense application

**INTERFACING DC MOTOR- 8051**

A DC motor runs in response to the applied Direct Current. It produces torque by using both electric and magnetic fields. The DC motor has rotor, stator, field magnet, brushes, shaft, commutator etc.. The DC motor requires large currents, of the order of 400mA for its rotation. But this much amount of current cannot be generated by the ports of the microcontroller. So, if you directly connect the DC motor to the ports of the controller, it may draw high current for its operation from the ports and hence the microcontroller may be damaged. Therefore we use a driving circuit along with an opto-isolator .The opto-isolator provides an additional protection to the microcontroller from large currents.

The interface circuit consists of L293D IC which is a dual H-bridge motor driver. This is a 16-pin chip, which can be used to interface two DC motors and can be controlled in both clockwise and anticlockwise direction of motion. L293 D has an output current of 600mA and peak output current of 1.2A per channel. Also for protection of the circuit from back emf, output diodes are also included in the IC. The output supply has a wide range from 4.5 V to 36V.

The ILQ74 is A Quad chip with four, optically coupled Gallium Arsenide infrared LED and silicon NPN phototransistor pairs. Signal information, including a DC level, can be transmitted by the device while maintaining a high degree of electrical isolation between input and output.



The interface circuit is build as shown in the figure. The port1 pins P1.0 ,P1.1 &P1.2 are connected to the opto-isolator inputs and the outputs of the opto-isolator are connected to the IN1,IN2 and the Enable1 pins of the L293D driver IC. When enable is High and EN2 is low and EN1 is high, the motor rotates in clockwise direction .Similarly if EN2 is high and EN1 is low with EN2 pin high, the DC motor rotates in anticlockwise direction. To rotate the motor the EN pin must be always high. Otherwise the status will be High impedance state. This can be shown by the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| Enable (EN) | Input1 | Input2 | Output |
| High | High | Low | Clockwise Rotation |
| High | Low | High | Anticlockwise |
| Low | Either High or Low | Either High or Low | High Impedance |

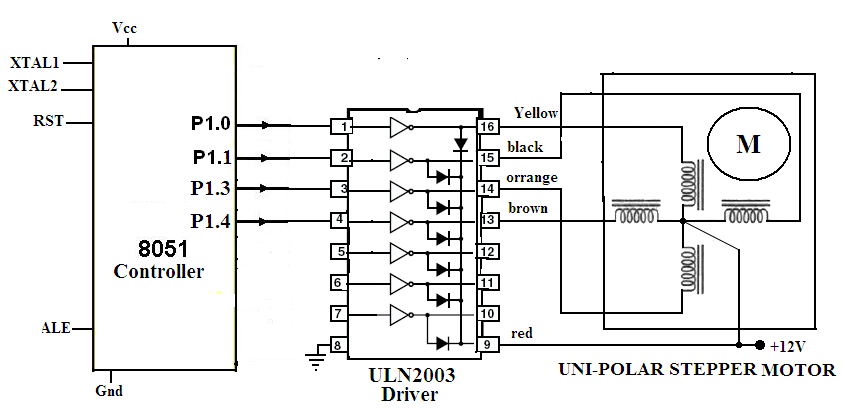
The rotation of the motor is controlled by writing a suitable program.

**Assembly Language program :**

|  |  |  |  |
| --- | --- | --- | --- |
| **ORG** | **0000H** |  | **Remarks** |
| MAIN | SETB | P1.2 | Enable the H-bridge driver |
|  | MOV | P1,#00000001B | Motor runs in clockwise |
|  | ACALL | DELAY |  |
|  | MOV | P1,#00000010B | Motor runs in anticlockwise |
|  | ACALL | DELAY |  |
|  | SJMP | MAIN | Motor rotates continuously in clockwise for some time and anticlockwise for some time |
| **DELAY** | MOV | R4,#FF H | Load R4 register with FF |
|  | MOV | R3,#FF H | Load R3 register with FF |
| LOOP1 | DJNZ | R3,LOOP1 | Decrement R3 until it is zero |
| LOOP2 | DJNZ | R4,LOOP2 | Decrement R4 until it is zero |
|  | RET |  | Return to the main program |

**Stepper motor Interfacing -8051**

A stepper motor is a device that translates electrical pulses into mechanical movement in steps of fixed step angle. So, The stepper motor rotates in steps in response to the applied signals. It is used in disk drives, dot matrix printers, plotters and robotics and process control industries. It is mainly used for position control. Stepper motors have a permanent magnet called rotor (also called the shaft) surrounded by a stator . There are also steppers called variable reluctance stepper motors that do not have a PM rotor. The most common stepper motors have four stator windings that are paired with a center-tap. This type of stepper motor is commonly referred to as a. four-phase or unipolar stepper motor. The center tap allows a change of current direction in each of two coils when a winding is grounded, thereby resulting in a polarity change of the stator.



Even a small stepper motor require a current of 400mA for its rotation. But the ports of the microcontroller can not source this much amount of current .If you try to connect the motor directly to the microcontroller ,the motor try to draw large current from the controller and the microcontroller will be damaged .So, we need to use a driver circuit ,to increase the port current of the microcontroller. This can be achieved by using an array of power transistors. But now a days we have driver circuits available readily in the form of ICs.ULN2003 is one such driver IC which is a High-Voltage High-Current Darlington transistor array and can give a current of 500mA. This current is sufficient to drive the stepper motor. This IC is also provided with diodes that can avoid the damage of motor due to back emf and large eddy currents. So, this ULN2003 is used as a driver to interface the stepper motor to the microcontroller.

The significant thing in a stepper motor is the step angle. It is the minimum angle through which the motor rotates in response to each excitation pulse. In a four phase motor if there are 200 steps in one complete rotation then then the step angle is 360/200 = 1.8 0. So to rotate the stepper motor we have to apply the excitation pulse. For this we send a hexa decimal code through the ports of the microcontroller. The hex code mainly depends on the construction of the steppe motor. So, all the stepper motors do not have the same Hex code for their rotation. We have to refer the operation manual supplied by the manufacturer.

For example , let us consider the hex code for a stepper motor to rotate in clockwise direction is 77H , BBH , DDH and EEH. This hex code will be applied to the input terminals of the driver through the assembly language program. To rotate the stepper motor in anti-clockwise direction the same code is applied in the reverse order. The assembly language program is given below.

**ASSEMBLY LANGUAGE PROGRAM**

Main : MOV A,# 0FF H ; Initialisation of Port 1

MOV P1, A ;

MOV A ,#77 H ; Code for the Phase 1

MOV P1 , A ;

ACALL DELAY ; Delay subroutine

MOV A ,# BB H ; Code for the Phase II

MOV P1 , A ;

ACALL DELAY ; Delay subroutine.

MOV A ,# DD H ; Code for the Phase III

MOV P1 , A ;

ACALL DELAY ; Delay subroutine

MOV A ,# EE H ; Code for the Phase 1

MOV P1 , A ;

ACALL DELAY ; Delay subroutine

SJMP MAIN ; Keep the motor rotating continuously.

**DELAY Subroutine**

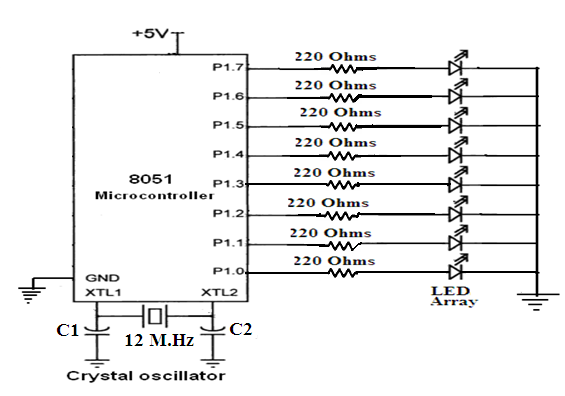
MOV R4 ,#0FF H ; Load R4 with FF

MOV R5,# 0FF ; Load R5 with FF

LOOP1: DJNZ R4 ,LOOP1 ; Decrement R4 until zero,wait

LOOP2: DJNZ R5 ,LOOP2 ; Decrement R5 until zero,wait

RET ; Return to main program .

**Binary Counter :** A binary counter is just like any other counter which counts the numbers in binary form. An 8-bit binary counter counts from 00000000 to 11111111.This binary counter can be realized by interfacing 8-LEDs to the port of the Microcontroller. By sending the continuous pulses through the port, the LEDs are glown continuosly one after the other. The glow of an LED indicates the binary 1 and the No Glow of LED denotes the binary 0.  ****

The interface circuit is shown in the above diagram. A array of LEDs is interfaced to the Port1 of the microcontroller. To drive the LEDs properly without any damage to the circuit, resistors of 220 ohms are connected in series with the LEDs as shown in the circuit. Suitable assembly program is used to realize the binary counter so that the LEDs glow from 00 to FF.

**Assembly language Program :**

START : MOV A, #00 ; load accumulator with 00

LOOP : MOV P1 , A ; send the contents of accumulator to Port1

CALL DELAY ; Create some delay

INC A ; Increment the content of accumulator.

CJNE A, # 0FF , LOOP ; compare the contents of A with FF and if not equal go to loop.

MOV P1, A ; send the value FF to Port 1

SJMP START ; keep the counter working continuously.

This program counts from 00 to FF and after reaching FF again it starts counting from 00.

**Note :** The binary counter circuit can also be designed using an inverter buffer IC with port.

**Interfacing LCD Module -8051 Microcontroller :**

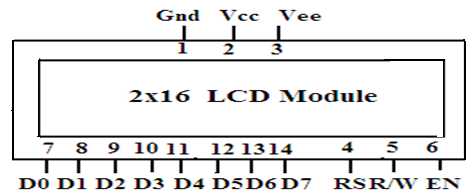
LCD modules are based on liquid crystaltechnology. The seven segment displays suffer from dissipation of power and also they are not suitable for display of graphs etc.. Hence the Liquid Crystal modules have become very popular now a days. They consume less power and they are highly suitable to display graphics, numbers and characters etc..

General LCD modules are available in different version like 1x16 (one line -16 characters) or 2x16 (two lines -16 characters) or 2 x 20(two lines 20 characters) etc.. The pin diagram of the LCD module is shown below. It is a 14-pin module in which 8 pins are for data, one pin is Vcc, one pin is Gnd ,one pin is for contrast(Vee) ,one pin is for RS, one pin is for EN and another pin is used for R/W.

The description of the pins is given below.

The pins 7 -1 4 denotes the data pins which are used to transfer Data .These pins are used to send information to the LCD or to read the contents of the LCD’s internal registers.

RS is the register select pin.. The LCD has two internal registers. One is command code register and the other is the data register. The command code register is used to send commands like clear display, cursor at home etc.. The data register allows the user to send the data to be displayed on the LCD.



When RS = 1, the data register is selected and when RS = 0 the command code register is selected.

RS = 0 is also used to check the busy flagbit to find whether the LCD is ready to receive information. The busy flag is D7 and can be read when R/W = 1 and RS=0.

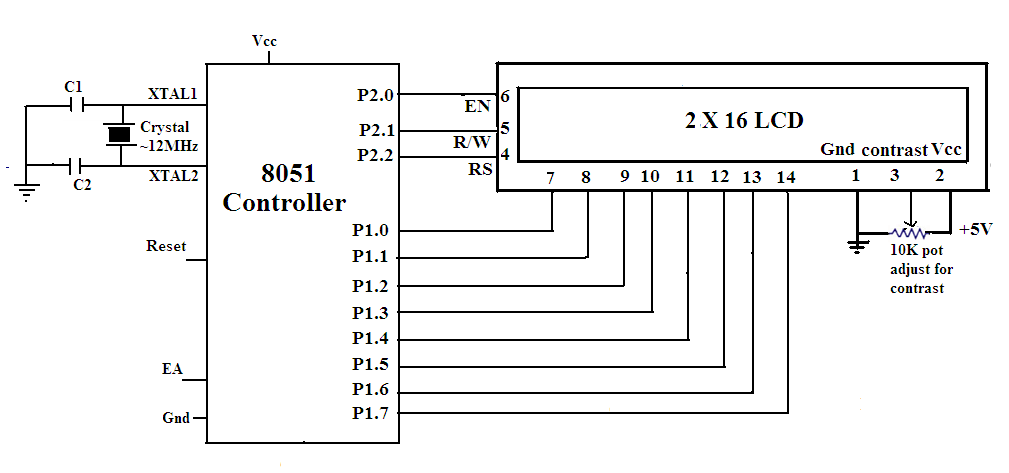
For ex: if R/W= 1 ,RS=0 AND D7 = 1 means the LCD is busy due to internal operations and will not accept any new data. Suppose D7=0 means the LCD is ready to receive the data.

R/W pin allows the user to read the information from LCD or write information to the LCD. When R/W = 0 writing is enabled and R/W=1 read is enabled.

EN pin is used by the LCD to latch information on its data pins.When data is given to data pins,a high to low pulse must be applied to this pin so that the LCD can Latch in the data present at the data pins.

Vcc ,Vss and Vee pins denote supply , ground and contrast pins respectively. Vee pin is meant for adjusting the contrast of the LCD display and the contrast can be adjusted by varying the voltage at this pin. This is done by connecting one end of a POT to the Vcc (5V), other end to the Ground and connecting the center terminal (wiper) of of the POT to the Vee pin. The contrast pin is applied a negative voltage so that by adjusting this voltage the contrast of the Display can be controlled.

**Interfacing circuit:** The interfacing of LCD module to 8051 microcontroller is shown below. The 8-data pins are connected to Port1 and control pins are connected to Port 2 pins. The busy flag must be checked before the data is send to the LCD. With the help of a 10K potentiometer the contrast of the LCD display is adjusted.



**The assembly language program to diplay “ELEC” on LCD is given below**

MOV A, #38H ; Use 2 lines and 5x7 matrix

ACALL COMMAND ; Issue a command

MOV A, #0FH : LCD ON, cursor ON, cursor blinking ON

ACALL COMMAND ; Command subroutine

MOV A, #01H ; Clear screen

ACALL COMMAND ; Command subroutine

MOV A,#06H ; Shift cursor right

ACALL COMMAND ; Command subroutine

MOV A , #80H ; Start display from Cursor line home

ACALL COMMAND ; Command subroutine

MOV A, # ’E’ ; Display letter E

ACALL DISPLAY ; Display subroutine

MOV A , # ‘L’ ; Display letter L

ACALL DISPLAY ; Display subroutine

MOV A, # ’E’ ; Display letter E

ACALL DISPLAY ; Display subroutine

MOV A, # ’C’ ; Display letter C

ACALL DISPLAY ; Display subroutine

HERE : SJMP HERE ;Display the data continuously

COMMAND SUBROUTINE :

MOV P1, A ; Issue command code .

CLR P2.2 ; RS = 0

CLR P2.1 ; R/W =0 to write to LCD

SETB P2.0 ; EN =1 pulse to enable

CLR P2.0 ; EN =0 latch in

RET

**DATA DISPLAY SUBROUTINE**

MOV P1, A ; Issue data

SETB P2.2 ; RS = 1 data register

CLR P2.1 ; R/W = 0 Write to LCD

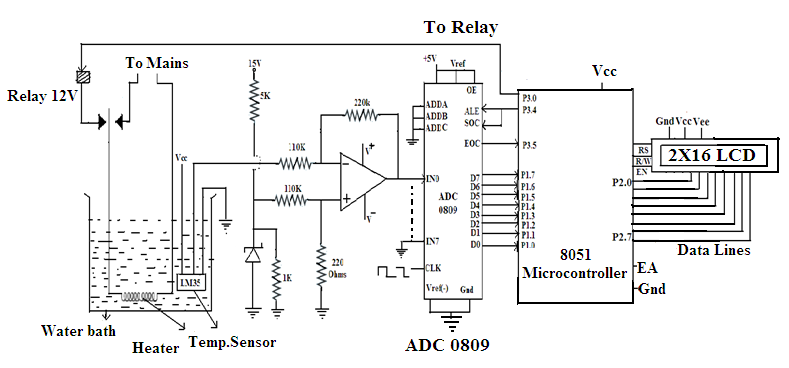
SETB P2.0 : EN = 1 for high to low pulse

CLR P2.0 ; EN = 0 Latch in

RET;

**Temperature measurement and controlling :** The control and measurement of temperature is always an important task.A microcontroller can control and measure the temperature effectively.The temperature sensor LM35 is used to measure the temperature in degree centigrade.LM35 sensor is a precision integrated circuit whose output voltage is linearly proportional to the centigrade temperature.This LM35 do not require any external calibration and it gives 10mV output for 1 degree centigrade.(10mV/0C).The LM35 has atypical 10C error over 1000C temperature range.

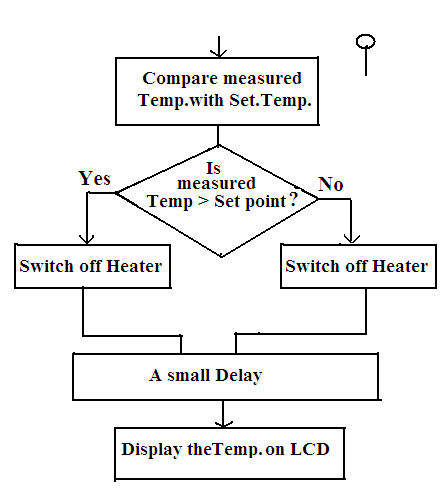
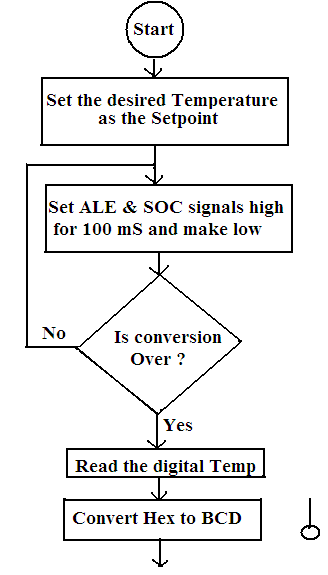
As shown in the circuit ADC0809 is used to covert the analog voltage into its equivalent digital voltage.For this necessary signal conditioning circuit is developed using an operational amplifier.The ADC0809 is an 8-input channel system.A clock signal of 100KHz is applied.the voltage drop across the LM Zener is buffered by an op-amp LM741to produce a Vcc and Vref of around 5V to 12Vfor the A/D converter.With this reference voltage the A/D converter will have 256 steps of 20mV each. Since the temperature sensor signal is amplified by 2 ,each degree Celsius of temperature change will produce an output change of 20mV.or one step on the A/D converter. This gives us a resolution of 10C ,which is about the typical accuracy of the sensor.The advantage of using Vref as Vcc for the device is that this voltage will not have the switching noise that the digital Vcc lines have.The control inputs and data outputs of the ADC are connected to the Port1 of the Microcontroller as shown in the circuit.



A 2x16 LCD module is interfaced to the Microcontroller to display the temperature .Suitable program is developed to control and measure the temperature. The temperature at which the water bath is to be maintained is set as (Ts).The measured temperature is compared with this Ts and if measured temperature is higher than the set temperature, a signal is sent to the relay to switch-off the power. Otherwise the power is switched on. This temperature is displayed on the LCD module.

The flow chart for the measurement of the Temperature is shown below.

**Flow Chart for Temperature measurement.**

****

**Assembly Language Program :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ddress(Hex)** | **Label** | **Mnemonics** | **Operands** | **Comments** |
| 9000 | START | SETB | P3.4 | Send ALE & SOC high |
|  |  | NOP |  | No operation |
|  |  | NOP |  | No operation |
|  |  | CLR | P3.4 | ALE &SOC low |
|  | LOOP1 | JNB | P3.5 , LOOP1 | Is conversion over ? |
|  |  | MOV | A,P1 | Read digital data |
|  |  | CLR | C | Clear carry bit |
|  |  | SUBB | A,#SET POINT | Compare the measured with Set temp. |
|  |  | JNC | OFF |  |
|  | ON | SETB | P3.0 | Switch on heater |
|  |  | CALL | CONV | Convert the hex value into Decimal value |
|  |  | CALL | DISPLAY | Display subroutine to display temp .on LCD |
|  | OFF | CLRB | P3.0 | Switch off heater |
|  |  | SJMP | START | Start sensing the temperature |
|  | DISPLAY | LCALL | 675F |  |
|  |  | MOV | R7, #00 |  |
|  |  | MOV | A, #Set point |  |
|  |  | MOV | R6,60H | Display set temp. |
|  |  | RET |  | Return to main program |

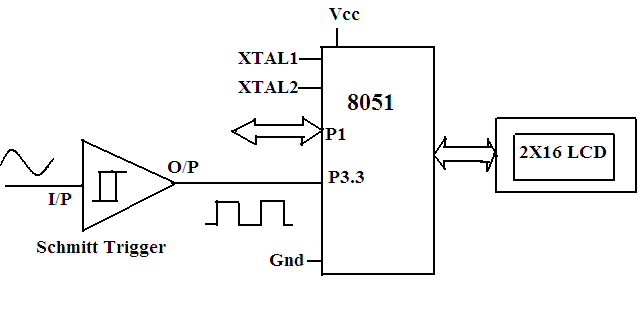
**PULSEWIDTH MEASUREMENT -8051**

The width of a pulse is equal to the sum of OFF & ON times. Normally a microcontroller is used to measure the OFF time or ON time of a perfect Square pulse .By doubling this value we get the total width of a perfect square pulse.

A perfect square pulse must be applied to INT0 or INT1 (P3.3 or P3.4) of the 8051 , Microcontroller. One of the Timers ,either T0 or T1 is used as timer to measure the pulse width.

In the present method the Timer0 is configured as a 16 bit timer in Mode 1 by using the TMOD register. The timer will be set once the pulse appears at the pin P3.3.The timer will run until the edge is high. Once the pulse edge goes low, the timer is stopped.The value of the timer0 is moved to registers R6 and R7 and later the total pulse width is determined..

A very important precaution to be followed here is that ,the applied pulse must be always a perfect square pulse. So, one can use a Schmitt trigger to convert a sine pulse into a square pulse and it is applied to the pin P3.3 or P3.4 as shown in figure below.



**Assembly Language Program :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Address** | **Lable** | **Instruction** | **Comments** |
|  | Main  Loop1  Loop2  Loop3 | MOV TMOD, #09  MOV TL0,#00  MOV TH0,#00  JB P3.3 ,Loop1  JNB P3.3 , Loop2  SETB TR0  JB P3.3 , Loop3  CLR TR0  MOV R6 ,TL0  MOV R7 TH0  LCALL 675F  SJMP Main | Initialize timer 0 in mode 1  Load Timer0 with 0000H.  Continue the loop as long as pulse is high  Start the timer 0  Clear the timer 0  Move the contents of Timer0 into R6 &R7 registers.  Call Display subroutine, to display the data.  Ready for next measurement. |

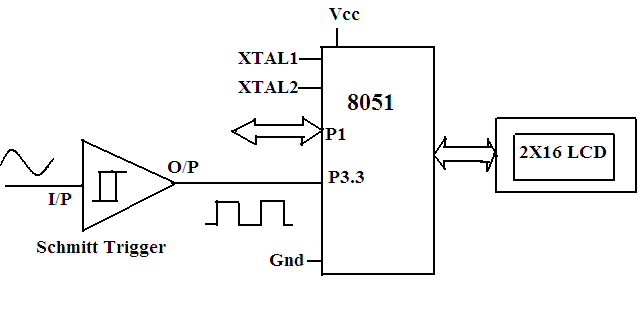
**FREQUENCY MEASUREMENT -8051 :**

Frequency of the given pulse train is equal to the number of pulses per second.

Frquency = Number of pulses / Time.

To find the frequency using the microcontroller, both the Timers T0 and T1 of 8051 are used. Normally if T0 is used as a counter ,the T1 is used as a Timer. The procedure is , the external pulse train is applied at the PinP3.3 or P3.4 . The Timer1 is configured to lapse for a known time . Say,100 micro seconds. During this time the Timer0 is used as a counter and it counts the total number of pulses appeared at the pinP3.3or P3.4.From this the number of pulses per second is calculated, which is nothing but the frequency of the pulse.

To measure the frequency, it is recommended to use a square wave pulses. If exact square wave is not available from a function generator , a Schimitt trigger circuit can be used ,which converts a sine wave into a square wave..The circuit diagram is shown below.



**Procedure :** In the procedure Timer0 is configured as a counter and the Timer1 is configured as a 16 bit timer. Timer 1 is pre-loaded with a suitable value so that it will run for time which generates a time lapse of 100 microseconds. During this time the Timer0 which is configured as a counter will count the number of pulses. The count value is divided by the time. This give the frequency of the pulse train applied externally at the pinP3.3

The maximum frequency measured by the microcontroller depends on the clock frequency .Normally the 8051 microcontroller can measure accurately those frequencies which do not exceed f/24, where f is clock frequency of the controller.

The assembly language program for the measurement of frequency is given below.

**Assembly Language Program:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Address** | **Label** | **Mnemonics** | **comments** |
| 8000 | START  LOOP2:  LOOP1: | MOV TCON,#00  SETB PSW.3  MOV R0,#00  MOV R1,#00  CLR PSW.3  MOV TMOD,#25  MOV TL1, #9C  MOV TH0,#00  CLR FLAG  MOV TCON, #50  MOV IE, #88  JBC 0F LOOP1  SJMP LOOP2    MOV R6, TL0  MOV R7, TH0  LCALL Display  SJMP START | Initialize TCON register  Initialize Banks 1  Initialize R0 Register  Initialize R1 Register  Return Bank 1  Timer 0 as a counter  Timer 1 as a counter  Start TL1 at 9CH  TH1 Overflowing 100 clocks  Start timer and counter  Enable global and T1 over flow interrupt  Load the count values into R6 & R7 registers  Display subroutine to display the data  Ready for the next Measurement. |

**MEASUREMENT OF CURRENT- 8051 CONTROLLER :**

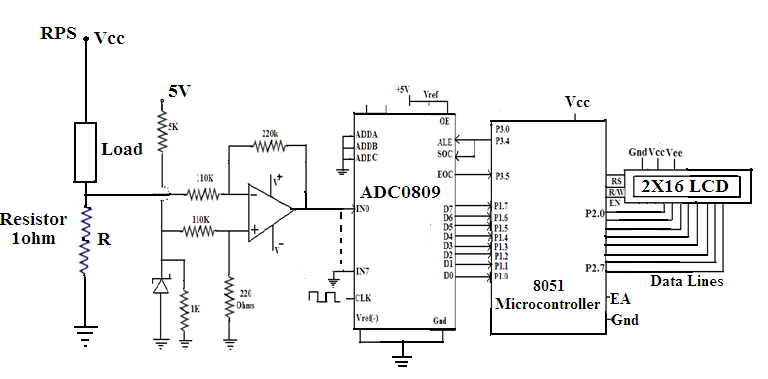
Small currents can be measured using the microcontroller. For this a resistor of small value between 0.1 to 1 ohm is connected to the load and Vcc in series as shown in the diagram.The voltage source must be always a regulated power supply. The voltage across the resistor is connected to the inverting terminal for signal conditioning. This will make the voltage suitable to apply to one of the inputs of the ADC0809.This ADC will convert this analog voltage into equivalent digital value.This digital signal is applied to the Port 1 of the microcontroller .The data is processed and displayed on the LCD module.

To calculate the current Ohm's law, V= IR , is used. In the above circuit the sense resistor 's value is 1 ohm and when 1 ampere of current is flowing in the resistor , then 1 Volt will be developed across the sense resistor. So, I = V / R

By measuring the voltage using the microcontroller, The current in the resistor can be found.

**Note :** This method is useful to measure small currents only. Because with large currents the small resistor will dissipate large heat. So, to measure large currents high wattage wire wound resistors or Torroid must be used.

The circuit diagram is shown below.



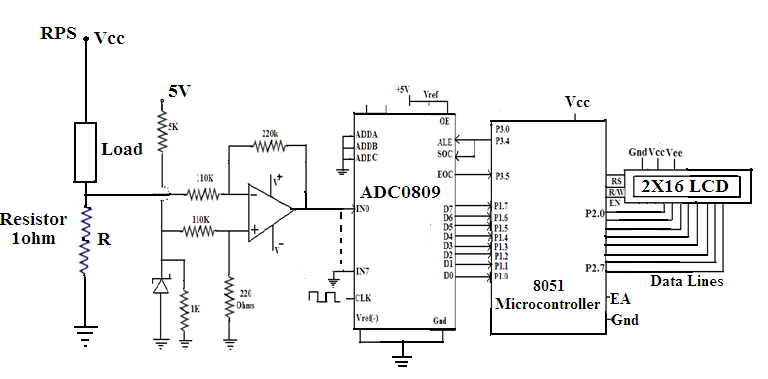
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address(Hex)** | **Label** | **Mnemonics** | **Operands** | **Comments** |
| 9000 | START | SETB | P3.4 | Send ALE & SOC high |
|  |  | NOP |  | No operation |
|  |  | NOP |  | No operation |
|  |  | CLR | P3.4 | ALE &SOC low |
|  | LOOP1 | JNB | P3.5 , LOOP1 | Is conversion over ? |
|  |  | MOV | A,P1 | Read digital data(Voltage) |
|  |  | CALL | CONV | Convert the hex value into Decimal value |
|  |  | MOV | B, # Resistor Value | Load the resistor value into B register |
|  |  | DIV | AB | Divide voltage by resistance |
|  |  | CALL | DISPLAY | Display subroutine to display current on LCD |
|  |  | SJMP | START | Measure the current continuously |

**Assembly Language Program** : The assembly language program is basically an ADC program

As given below.

**MEASUREMENT OF VOLTAGE- 8051 CONTROLLER :**

Small DC Voltage s can be measured using the microcontroller. For this a resistor of small value around 1 ohm is connected to the load and Vcc in series as shown in the diagram. The voltage source must be always a regulated power supply. The voltage across the resistor is connected to the inverting terminal for signal conditioning. This will make the voltage suitable to apply to one of the inputs of the ADC0809.This ADC will convert this analog voltage into equivalent digital value. This digital signal is applied to the Port 1 of the microcontroller .The data is processed and displayed on the LCD module.



**Note :** This method is useful to measure small currents only. Because with large currents the small resistor will dissipate large heat. So, to measure large voltages high wattage wire wound resistors or Torroid must be used.

**Assembly Language Program** : The assembly language program is basically an ADC program

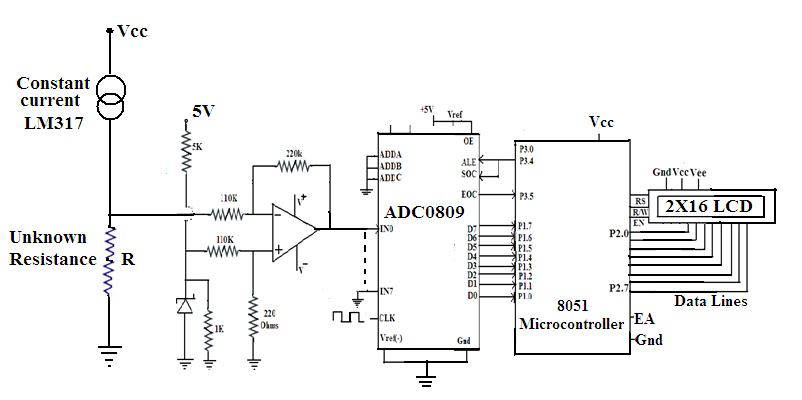
As given below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address(Hex)** | **Label** | **Mnemonics** | **Operands** | **Comments** |
| 9000 | START | SETB | P3.4 | Send ALE & SOC high |
|  |  | NOP |  | No operation |
|  |  | NOP |  | No operation |
|  |  | CLR | P3.4 | ALE &SOC low |
|  | LOOP1 | JNB | P3.5 , LOOP1 | Is conversion over ? |
|  |  | MOV | A,P1 | Read digital data(Voltage) |
|  |  | CALL | CONV | Convert the hex value into Decimal value |
|  |  | CALL | DISPLAY | Display subroutine to display Voltage on LCD |
|  |  | SJMP | START | Measure the voltage continuously |

**MEASUREMENT OF RESISTANCE - 8051 CONTROLLER**

Resistance is measured using the current-voltage relation. i.e using the Ohm’s law.In the relation V= IR if the voltage and current are known ,the resistance can be determined. In this method a constant current source is used. A resistor is connected in series to the current source .The voltage drop across the resistor is signal conditioned using the op-amp circuit and this analog signal is then converted into equivalent digital signal by using ADC. The digital signal is fed to the Port1 of the microcontroller. The microcontroller will process the data .Using the relation R=V/I , the resistor value is determined and by developing a suitable program the resistance value is displayed on the LCD. In the circuit LM317 regulator can be used as constant current source.

The interface diagram is shown below.



An assembly language program is developed to find the resistance value.The program is given below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address(Hex)** | **Label** | **Mnemonics** | **Operands** | **Comments** |
| 9000 | START | SETB | P3.4 | Send ALE & SOC high |
|  |  | NOP |  | No operation |
|  |  | NOP |  | No operation |
|  |  | CLR | P3.4 | ALE &SOC low |
|  | LOOP1 | JNB | P3.5 , LOOP1 | Is conversion over ? |
|  |  | MOV | A,P1 | Read digital data(Voltage) |
|  |  | CALL | CONV | Convert the hex value into Decimal value |
|  |  | MOV | B, # Current value | Load the known value of the current into B register |
|  |  | DIV | AB | Divide voltage by current |
|  |  | CALL | DISPLAY | Display subroutine to display resistance value on LCD |
|  |  | SJMP | START | Measure the resistance continuously |