

Aim:

To design a working model of a contactless digital tachometer using 8051 Microcontroller.

A Tachometer is a device which measures the speed of a rotating object like an electric motor or a crank shaft of a vehicle engine. Speed of an electric motor is determined by the number of revolutions made by the motor in one minute. In other words, speed is measured in RPM (Revolutions per Minute).

Principle :

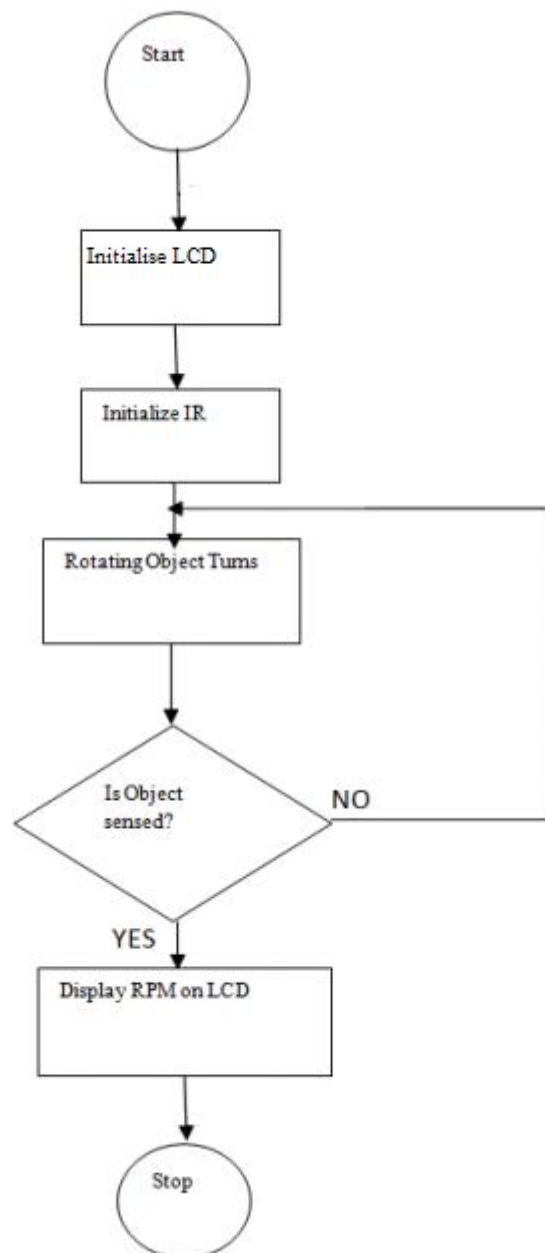
The basic principle behind the Contactless Digital Tachometer involves a simple embedded system with a sensor, a controller and an actuator. The sensor used here is Infrared (IR) transmitter – receiver pair, the controller used is the 8051 Microcontroller loaded with a compiled code and the actuator is a display device, for displaying the speed of the motor.

The sensor senses the speed of the motor without actually being in contact with it by the principle of light transmission and reflection and generates a signal. This signal is converted into an electric signal and fed to the microcontroller, which is programmed to calculate the speed in terms of number of motor revolutions in one minute. This speed is displayed on the LCD display.

Algorithm:

1. Start
2. Initialise LCD and timing delay function.
3. Initialise interrupt functions.
4. Delay function is called, and rotating object is detected by IR sensor.
5. Count variable is incremented upon detection of falling edge of signal generated by IR sensor.
6. Count is multiplied by 60 to obtain the required RPM.
7. The obtained RPM value is displayed on the LCD
8. Go to step 4

Flowchart:



Code:

```
#include<reg51.h>

#include<string.h>

#include<stdio.h>

unsigned char str1[10]="RPM= ";
unsigned char str2[5];
unsigned char temp[10];
unsigned char cmd[5]={0x38,0x0E,0x06,0x01,0x80};
unsigned char i;

sbit E=P0^0;  //enable pin
sbit RW=P0^1;      //read/write pin
sbit RS=P0^2; //register select pin
sbit busy= P1^7; //busy pin

typedef unsigned int uint;

uint count = -1,rpm=0;

void delay(uint x)
{
    int i;
    for(i=0;i<x;i++)
    {
        TMOD = 0x01;      // Timer 0 Mode 1
        TH0= 0xFC;          //initial value for 1ms
        TL0 = 0x66;
        TR0 = 1;           // timer start
        while (TF0 == 0);  // check overflow condition
```

```

        TR0 = 0;           // Stop Timer
        TF0 = 0;           // Clear flag
    }
}

```

void lcdready()//to prepare LCD and get it ready

```

{
    busy=1; // to set as input line
    RS=0;
    RW=1;
    while(busy==1)
    {
        //reading pin
        E=0;
        delay(1);
        E=1;
    }
} // end of lcdready

```

void cmdwrt(unsigned char cmdval)//to send command instruction to LCD

```

{
    lcdready();
    P1=cmdval;
    RS=0; //to write command in command register
    RW=0;
    E=1;  //enable=1, to enable LCD and take data from data lines
    delay(1);
    E=0;
} // end of cmdwrt

```

```
void lcddata(unsigned char dataval)// to send display data to LCD
```

```
{  
    lcdready();  
    P1=dataval;  
    RS=1;  
    RW=0;  
    E=1;  
    delay(1);  
    E=0;
```

```
}// end of lcddata
```

```
void int_to_str(unsigned char str[],unsigned int x)// to convert rpm value from int to string
```

```
{  
    int i, rem, len = 0, n;  
  
    n = x;  
    while (n != 0)  
    {  
        len++;  
        n /= 10;  
    }  
    for (i = 0; i < len; i++)  
    {  
        rem = x % 10;  
        x = x / 10;  
        str[len - (i + 1)] = rem + '0';  
    }  
}
```

```
str[len] = '\0';
```

```
}// end of int_to_str
```

```
void Int_Init()
```

```
{
```

```
    IT0 = 1;           // set detection to falling edge (sets external interrupt event type  
onto int0 pin)
```

```
    EX0 = 1;          // Enable external interrupt or int0 or p3.2 pin
```

```
    EA = 1;           //Enable global interrupt
```

```
}
```

```
void ex0_ISR(void) interrupt 0      // interrupt service routine
```

```
{
```

```
    count++;
```

```
    //delay(500);
```

```
}
```

```
void lcdinit() // to initialise the LCD
```

```
{
```

```
    unsigned char x;
```

```
    for(x=0;x<5;x++)
```

```
        cmdwrt(cmd[x]);
```

```
}
```

```

void main()
{
    Int_Init();// external interrupt initialize function
    lcdinit();

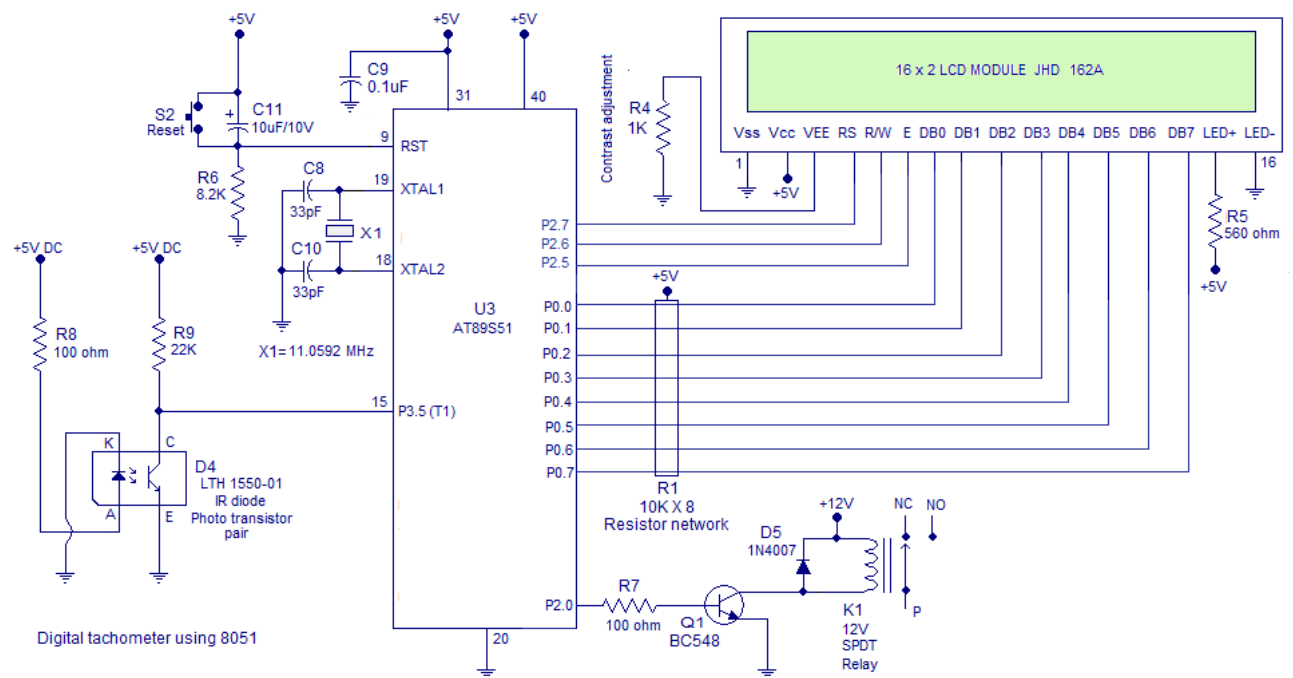
    while(1)
    {
        rpm=0;
        count=0;
        delay(1000);
        lcdinit();
        cmdwrt(0x01);// to clear the screen

        rpm = count * 60;

        int_to_str(str2,rpm); // convert int to string
        strcpy(temp,str1);
        strcat(temp,str2);    // concatenate the rpm value string to original string
        for(i=0;i<strlen(temp);i++)    // sending data to function to display on LCD
            lcddata(temp[i]);
    }
}

```

Circuit Diagram:



Result:

A working model of contactless digital tachometer was successfully designed using 8051 Microcontroller.