**DIGITAL TACHOMETER**

Course Project Report for

**Embedded System Project (EET 4100)**

**Submitted by**

|  |  |
| --- | --- |
| KISHAN KUMAR SINGH | 1641016394 |
| TASHMIN MISHRA | 1641016433 |
| FEECON BEHERA | 1641016437 |
| SUBHAM KUMAR | 1641016486 |



**DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING**

Institute of Technical Education and Research

(Faculty of Engineering)

**SIKSHA ‘O’ ANUSANDHAN DEEMED TO BE UNIVERSITY**

Bhubaneswar, Odisha, India

(November, 2019)

# DECLARATION

We, the undersigned students of B. Tech. (7th Semester) of Electronics and Communication Engineering program hereby declare that we own the full responsibility for the information, results etc. provided in this PROJECT titled “**DIGITAL TACHOMETER**” submitted to **Siksha ‘O’ Anusandhan Deemed to be University, Bhubaneswar** for the partial fulfillment of the subject **Embedded System Project (EET 4100)**. We have taken care in all respect to honor the intellectual property right and have acknowledged the contribution of others for using them in academic purpose and further declare that in case of any violation of intellectual property right or copyright we, as the candidates, will be fully responsible for the same.

|  |  |  |
| --- | --- | --- |
| **KISHAN KUMAR SINGH**  Registration No.:16410164394 |  | **TASHMIN MISHRA**  Registration No.:1641016433 |
| **FEECON BEHERA**  Registration No.:1641016437 |  | **SUBHAM KUMAR**  Registration No.:1641016486 |

**DATE: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**PLACE: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# ABSTRACT

A digital tachometer is a digital device that measures and indicates the speed of a rotating object. A rotating object may be a bike tyre, a car tyre or a ceiling fan, or [any other motor](https://www.edgefxkits.com/closed-loop-control-for-a-brushless-dc-motor-to-run-at-the-exactly-entered-speed), and so on. A digital tachometer circuit comprises LCD or [LED](https://www.elprocus.com/led-light-sources/) read out and a memory for storage. Digital tachometers are more common these days and they provide numerical readings instead of dials and needles.

# CONTENTS

[DECLARATION1](#_Toc26497797)

[ABSTRACT2](#_Toc26497798)

[CONTENTS3](#_Toc26497799)

[1. INTRODUCTION1](#_Toc26497800)

[1.1. Introduction1](#_Toc26497801)

[1.2. Problem Statement1](#_Toc26497802)

[1.3. Organization of the Report1](#_Toc26497803)

[2. DESIGN & IMPLEMENTATION2](#_Toc26497804)

[2.1. Hardware Design2](#_Toc26497805)

[2.2. Hardware Interface3](#_Toc26497806)

[2.3. Software Design4](#_Toc26497807)

[2.4. Implementation5](#_Toc26497808)

[3. TESTING, RESULTS AND DISCUSSION6](#_Toc26497809)

[4. PROBLEMS & CHALLENGES FACED8](#_Toc26497810)

[4.1. Bug/issue 018](#_Toc26497811)

[4.2. Bug/issue 028](#_Toc26497812)

[4.3. Bug/issue 038](#_Toc26497813)

[5. CONCLUSIONS AND RECOMMENDATIONS9](#_Toc26497814)

[6. REFERENCES10](#_Toc26497815)

[APPENDIX - I11](#_Toc26497816)

[Team Members & Responsibilities11](#_Toc26497817)

[Schedule11](#_Toc26497818)

[Parts List & Cost11](#_Toc26497819)

[APPENDIX - II11](#_Toc26497820)

[APPENDIX - III17](#_Toc26497821)

# INTRODUCTION

## 1.1. Introduction

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). Here, in this project, we designed a simple digital Tachometer using 8051 Microcontroller, which can measure speed with an accuracy of 1 rev/sec. A three digit contact less digital tachometer using 8051 microcontroller which can be used for measuring the revolutions/second of a rotating wheel, disc, shaft or anything like that is introduced in this project. The tachometer can measure up to a maximum of 255 rev/sec at an accuracy of 1 rev/sec.

## 1.2. Problem Statement

To design a digital tachometer using 8051 microcontroller.

## 1.3. Organization of the Report

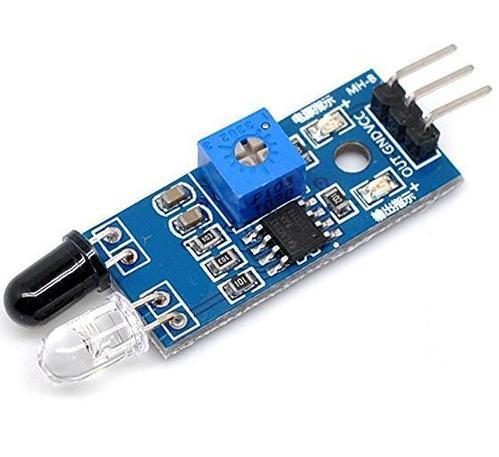
# DESIGN & IMPLEMENTATION

## 2.1. Hardware Design

This project consist of three units:-

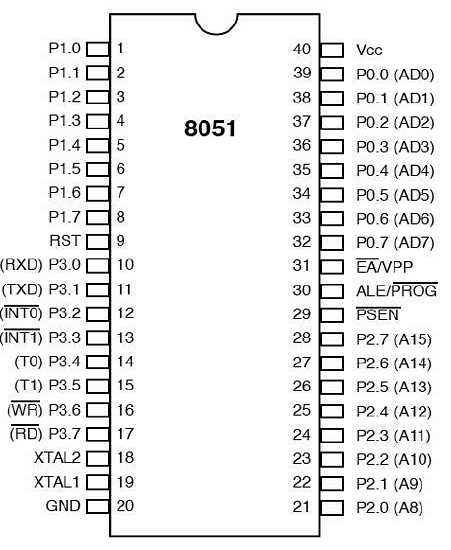
* Sensing unit
* Controlling unit
* Displaying unit

**Sensing unit**

Sensor Module has a pair of infrared transmitting and receiving tubes. When the transmitted light waves are reflected back, the reflected IR waves will be received by the receiver tube. The onboard comparator circuitry does the processing and the green indicator LED comes to life.

The module features a 3 wire interface with Vcc, GND and an OUTPUT pin on its tail. It works fine with 3.3 to 5V levels. Upon hindrance/reflectance, the output pin gives out a digital signal (a low-level signal). The onboard preset helps to fine tune the range of operation, effective distance range is 2cm to 80cm.

**Controlling unit**

8051 microcontroller is designed by Intel in 1981. It is an 8-bit microcontroller. It is built with 40 pins DIP (dual inline package), 4kb of ROM storage and 128 bytes of RAM storage, two 16-bit timers. It consists of are four parallel 8-bit ports, which are programmable as well as addressable as per the requirement.

There are two buses in 8051 Microcontroller one for program and other for data. As a result, it has two storage rooms for both program and data of 64K by 8 size. The microcontroller comprise of 8 bit accumulator & 8 bit processing unit. It also consists of 8 bit B register as majorly functioning blocks and 8051 microcontroller programming is done with [embedded C language](https://www.elprocus.com/basics-and-structure-of-embedded-c-program-with-examples-for-beginners/) using Keil software. It also has a number of other 8 bit and 16 bit registers.

The oscillating frequency of the microcontroller is 0.9216MHz, so for getting its oscillating frequency, we selected the values of ceramic capacitors to be 33pF and a quad crystal of value 11.0592 MHz as we are not using any external memory.

**Displaying unit**

The term [LCD stands for liquid crystal display](https://www.elprocus.com/difference-alphanumeric-display-and-customized-lcd/). It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc.

Here we are using a 16x2 LCD display. The 16×2 LCD display is a very basic module commonly used in [DIYs](https://electronicsforu.com/category/electronics-projects/hardware-diy) and circuits. The 16×2 translates a display 16 characters per line in 2 such lines. In this LCD each character is displayed in a 5×7 pixel matrix.

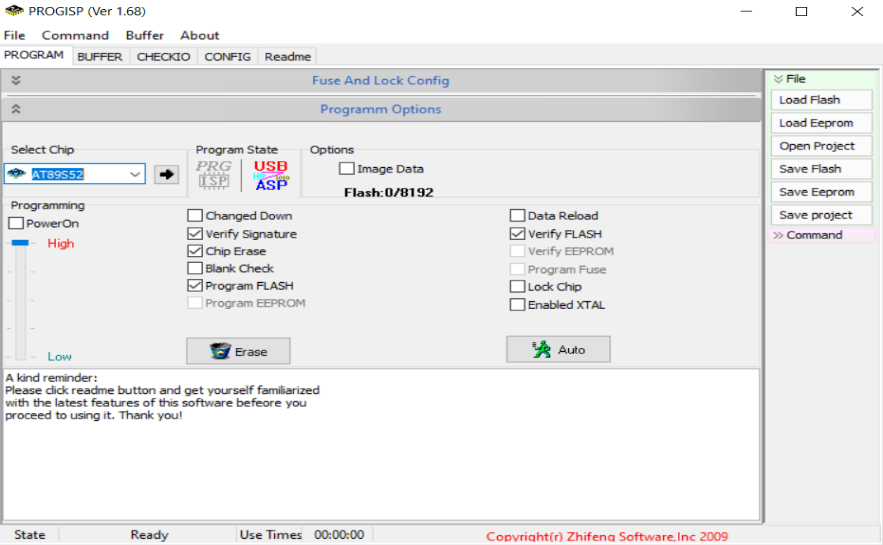
## 2.2. Hardware Interface

**8051 ISP Programmer**

A driver/programmer is needed to train the microcontroller that is to upload or burn the code which contains instructions to carry out the process, in order to drive the whole circuit properly we used a USB programmer to program the 8051 microcontroller which allows you to read or write the microcontroller flash, EEPROM, fuse bit and lock bits.

It has 6 pin out which are VCC, GND, MOSI, MISO, SCK, RESET which need to be connected through jumper wire with pin 40,20,6,7,8,9 of 8051 respectively in order to program the microcontroller. From here the hex file need to burn to the 8051 through PROGISP application.

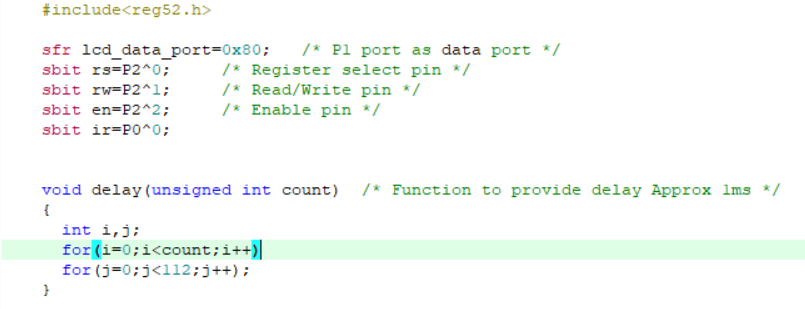
What need to do is to just select the chip from left corner for example AT80S52 and then go to Load Flash in right corner there you browse the hex file and lastly click on Auto to upload the code into 8051.

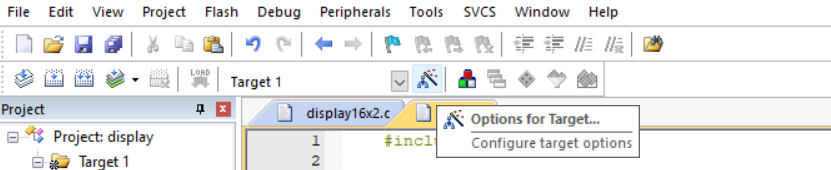


PROGISP application to burn the hex file.

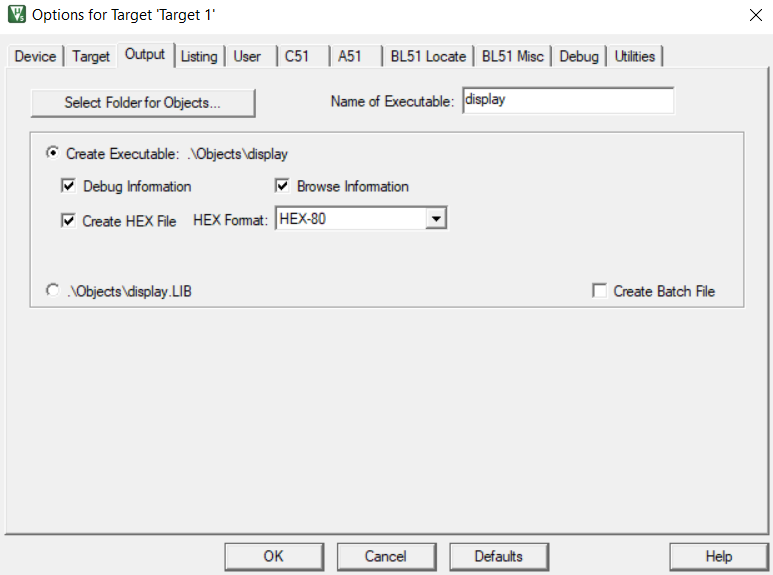
## 2.3. Software Design

In order to train the microcontroller to do a specific task we need a trainer that is to write a code which will instruct the microcontroller to the task according to the instruction. For code Keil uVision5 is used to write the code in C language. To write the code for a specific microcontroller we need to select proper chip to write the code for example AT89S52. After writing the code you need to create a HEX file to burn it on microcontroller.

Writing a code:

Creating a Hex file:

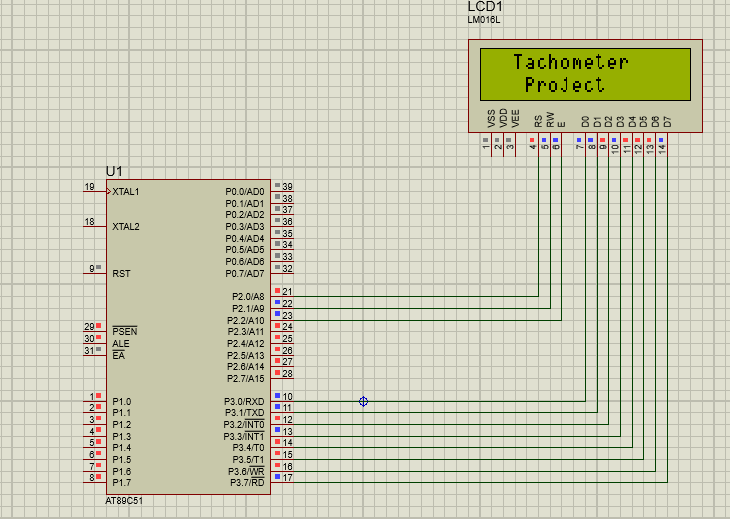
Just mode to Option for Target and then to output and check mark the create hex code to generate Hex code and code will save in same directory as the project is.



Creating a Hex file

For simulation purpose an application called Proteus is used to simulate the whole circuit. Before hardware implementation, a simulation is must to verify the circuit whether it is working or not. Here we used Proteus 8 to simulate the circuit.

16×2 display with 8051 microcontroller.



## 2.4. Implementation

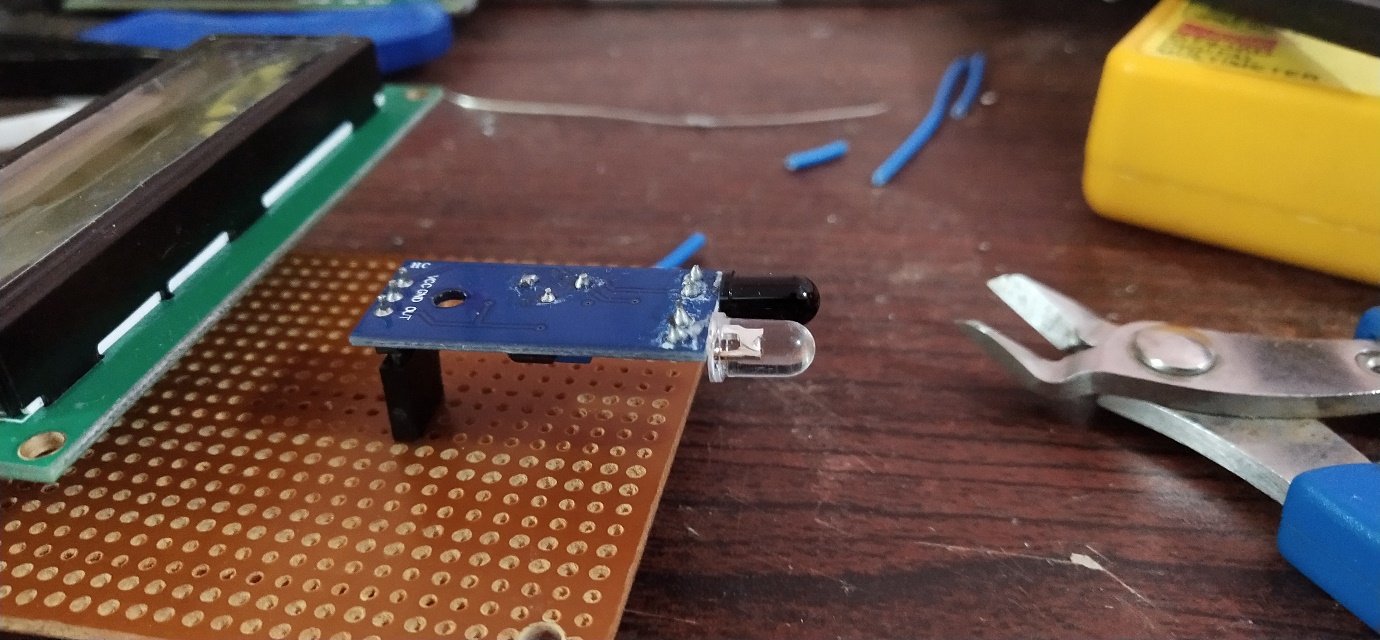


**Implementing Microcontroller**

Some RC circuit, Oscillator need to be implemented with the microcontroller to enable it in working situation. We have used AT89S52 microcontroller with 11.0592 MHz oscillator and 10kOhm resistor in reset pin along with a switch to reset it every time needed. Four header pin is out for MOSI, MISO, SCK and RESET pin for program uploading purpose.

**Implementing IR sensor:**

Here it is the sensing unit and will act as an input to the microcontroller giving information about the detection of white/black patch on the rotor. This sensor contain both transmitter and receiver. The transparent LED is the transmitter and the black one is the photodiode which is the receiver. It is connected to port0^1, whenever a white surface comes in front it reflect back the signal and it sends a high voltage to the microcontroller.



**Implementing Display**

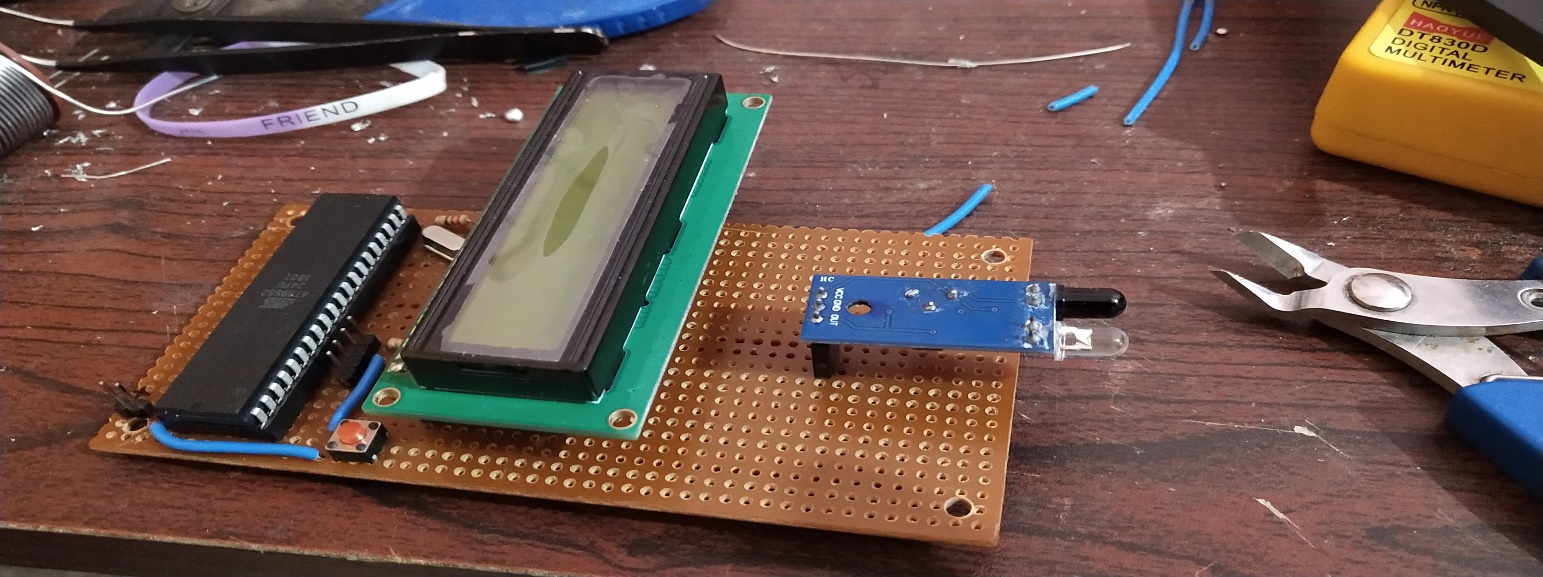
To display the data that is the rpm of the rotor we need a display to see the output data from microcontroller. Here we have used 16×2 display for the output purpose having 16pins out of which 8 pins are diode pin which is connected to port3 of microcontroller. To adjust the contrast we used a 220ohm resistor to get a better contrast on display.



**Final Implementation**



fvvv

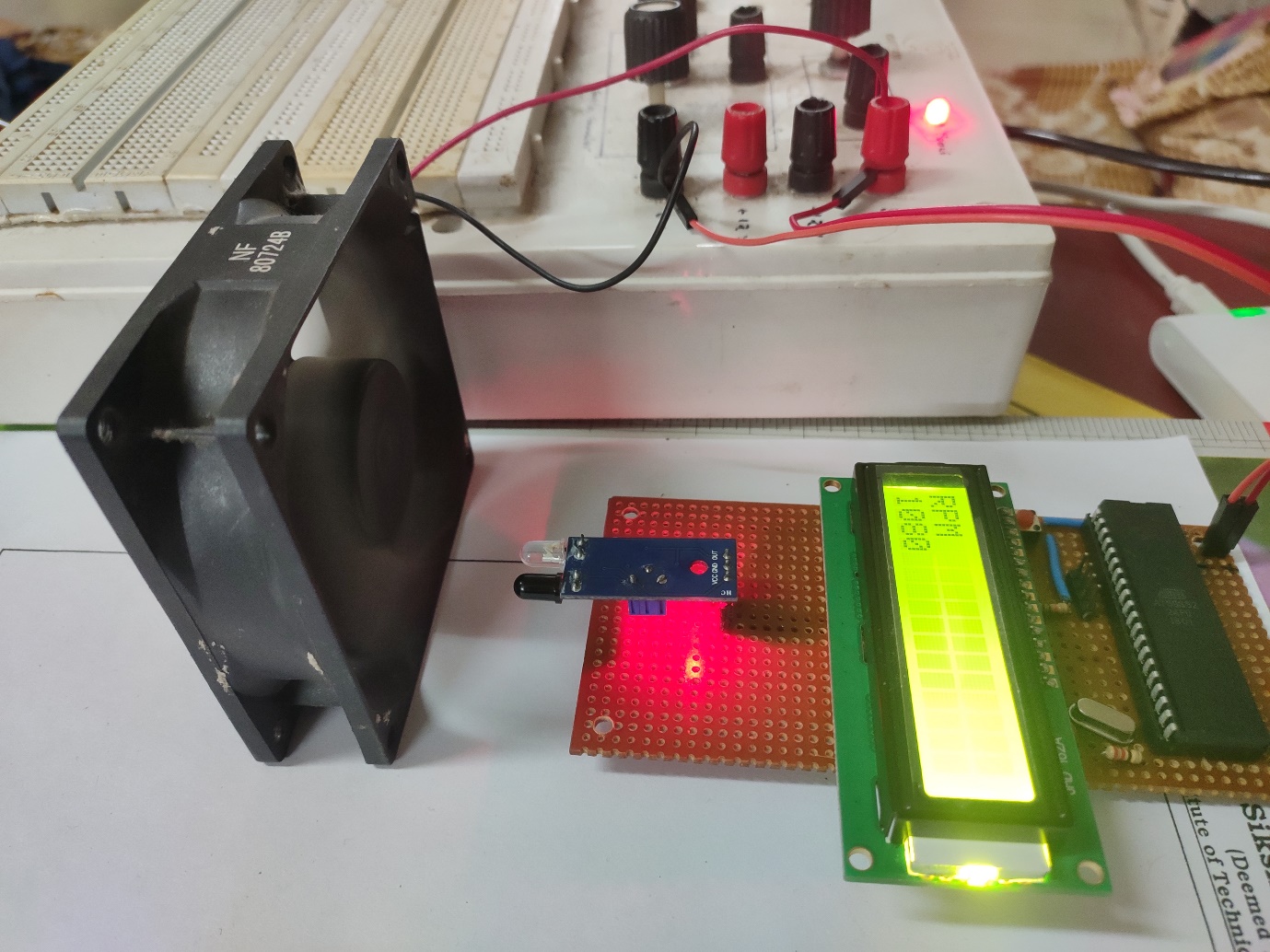


# TESTING, RESULTS AND DISCUSSION

For testing purpose a rotating fan is used where a white spot is there with black background so that the IR sensor can detect that and send a high value at that particular spot. The fan is faced against the sensor for the operation that is to measure the RPM.

This experiment was repeated more than once to check for consistency and errors and accordingly changes were made.

After continued testing and error correction we got the RPM to be around 1080 which was being successfully displayed on the screen.



# PROBLEMS & CHALLENGES FACED

## 4.1. Bug/issue 01

* **Chip Enable Error –** This is the error we were getting while uploading any program to AT89S552.Later we fixed this by pulling down the RESET pin of the microcontroller to GND using a 10K resistor.

## 4.2. Bug/issue 02

* **Too Much Contrast in LCD Display –** Initially it was very difficult to distinguish the characters displayed on the LCD which was successfully resolved to an acceptable level by putting a 5K resistor in between GND and Pin 3 of the LCD.

## 4.3. Bug/issue 03

* **TimeKeeping** - Basically to measure the RPM we had to find the duration between two continuous HIGH signals sent by the IR sensor and then calculate the RPM but using Timers wasn’t easy and the time duration was really small so what we did was we took duration over a constant period of time and also kept track of number of times the signal changed which at the end yielded the RPM.

# CONCLUSIONS AND RECOMMENDATIONS

So at last we can conclude that this Digital Tachometer works and achieves it’s goal of measuring RPM i.e. Revolutions Per Minute.

But we can improve this by getting a faster microcontroller as we suspect it might not work for faster microcontrollers. Also now it doesn’t measure the RPM on the fly but can will be an improvement and make it dynamic.

# REFERENCES

* [1] Shirriff, Ken (30 August 2016). ["The Surprising Story of the First Microprocessors"](https://spectrum.ieee.org/tech-history/silicon-revolution/the-surprising-story-of-the-first-microprocessors). [IEEE Spectrum](https://en.wikipedia.org/wiki/IEEE_Spectrum). [Institute of Electrical and Electronics Engineers](https://en.wikipedia.org/wiki/Institute_of_Electrical_and_Electronics_Engineers). Retrieved 13 October 2019.
* [2] Heath, Steve (2003). [Embedded systems design](https://books.google.com/books?id=BjNZXwH7HlkC&pg=PA11). EDN series for design engineers (2 ed.). Newnes. pp. 11–12. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [9780750655460](https://en.wikipedia.org/wiki/Special:BookSources/9780750655460).
* [3] Erjavec, Jack (2005). Automotive Technology. [ISBN](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [1-4018-4831-1](https://en.wikipedia.org/wiki/Special:BookSources/1-4018-4831-1).
* [[4] "Tachometer - Facts from the Encyclopedia - Yahoo! Education"](https://web.archive.org/web/20121106222358/http:/education.yahoo.com/reference/encyclopedia/entry/tachomet). Education.yahoo.com. Archived from [the original](http://education.yahoo.com/reference/encyclopedia/entry/tachomet) on 2012-11-06. Retrieved 2012-06-05.

# APPENDIX – I

## Team Members & Responsibilities

|  |  |  |
| --- | --- | --- |
| **Member** | **Responsibilities Undertaken** | **Level of performance**  **(Excellent/ Good/ Average)** |
| KISHAN KUMAR SINGH | PPT making | Good |
| TASHMIN MISHRA | Hardware Implementation | Good |
| FEECON BEHERA | Report Making | Good |
| SUBHAM KUMAR | Code Implementation | Good |
|  |  |  |

## Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **Week#** | **Date** | **Task** | **Actual** |
| 1 | 04/11/19 | PROJECT DESIGN | Completed |
| 2 | 11/11/19 | SOFTWARE PART/PROGRAMMING PART | Problems Encountered |
| 3 | 18/11/19 | HARDWARE IMPLEMENTATION | Completed |
| 4 | 25/11/19 | FINAL TESTING | Completed |

## Parts List & Cost

|  |  |  |
| --- | --- | --- |
| **Part No.** | **Part Name** | **Cost (in ₹)** |
|  | AT89S52 MICROCONTROLLER | 80 |
|  | MICROCONTROLLER BASE | 4 |
|  | 16X2 LCD DISPLAY | 130 |
|  | 33 PF CAPACITOR | 4 |
|  | 11.0592 MHZ CRYSTAL | 20 |
|  | JUMPER WIRE | 120 |
|  | PCB BOARD | 35 |
|  | IR SENSOR | 80 |
|  | Female Header | 10 |
|  | Male Header | 8 |
|  | **Total** | 371 |

# APPENDIX – II

#include<reg52.h>

sfr lcd\_data\_port=0xB0; /\* P1 port as data port \*/

sbit rs=P2^0; /\* Register select pin \*/

sbit rw=P2^1; /\* Read/Write pin \*/

sbit en=P2^2; /\* Enable pin \*/

sbit ir=P1^0;

unsigned char c;

int num[10];

void delay(unsigned int count) /\* Function to provide delay Approx 1ms \*/

{

int i, j;

for(i=0;i<count; i++)

for(j=0;j<112;j++);

}

void LCD\_Command (unsigned char cmd) /\* LCD16×2 command funtion \*/

{

lcd\_data\_port= cmd;

rs=0; /\* command reg. \*/

rw=0; /\* Write operation \*/

en=1;

delay(1);

en=0;

delay(5);

}

void LCD\_Char (unsigned char char\_data) /\* LCD data write function \*/

{

lcd\_data\_port=char\_data;

rs=1; /\* Data reg.\*/

rw=0; /\* Write operation\*/

en=1;

delay(1);

en=0;

delay(5);

}

void lcd\_data(unsigned int i) //Function to send data on LCD

{ int p;

int k=0;

while(i>0)

{

num[k]=i%10;

i=i/10;

k++;

}

k--;

for (p=k; p>=0;p--)

{

c=num[p]+48;

lcd\_data\_port = c;

rw = 0;

rs = 1;

en = 1;

delay(1);

en = 0;

}

return;

}

void LCD\_String (unsigned char \*str) /\* Send string to LCD function \*/

{

int i;

for(i=0;str[i]!=0;i++) /\* Send each char of string till the NULL \*/

{

LCD\_Char (str[i]); /\* Call LCD data write \*/

}

}

void LCD\_String\_xy (char row, char pos, char \*str) /\* Send string to LCD function \*/

{

if (row == 0)

LCD\_Command((pos & 0x0F)|0x80);

else if (row == 1)

LCD\_Command((pos & 0x0F)|0xC0);

LCD\_String(str); /\* Call LCD string function \*/

}

void LCD\_Init (void) /\* LCD Initialize function \*/

{

delay(20); /\* LCD Power ON Initialization time >15ms \*/

LCD\_Command (0×38); /\* Initialization of 16X2 LCD in 8bit mode \*/

LCD\_Command (0×0C); /\* Display ON Cursor OFF \*/

LCD\_Command (0×06); /\* Auto Increment cursor \*/

LCD\_Command (0×01); /\* clear display \*/

LCD\_Command (0×80); /\* cursor at home position \*/

}

void welcome()

{

LCD\_Init(); /\* initialization of LCD\*/

LCD\_Command(0×0C);

delay(20);

LCD\_String(" Tachometer"); /\* write string on 1st line of LCD\*/

delay(2000);

LCD\_Command(0xc0);

LCD\_String(" Project"); /\*write string on 2nd line\*/

delay(1000);

LCD\_Command(0×01);

LCD\_Command(0×0c);

LCD\_String("By-");

LCD\_Command(0xc0);

LCD\_String(" Tashmin Mishra");

delay(1000);

LCD\_Command(0×01);

LCD\_Command(0×0c);

LCD\_String("By-");

LCD\_Command(0xc0);

LCD\_String(" Feecon Behera");

delay(1000);

LCD\_Command(0×01);

LCD\_Command(0×0c);

LCD\_String("By-");

LCD\_Command(0xc0);

LCD\_String("Kishan kumar Singh");

delay(1000);

LCD\_Command(0x01);

LCD\_Command(0x0c);

LCD\_String("By-");

LCD\_Command(0xc0);

LCD\_String(" Subham Kumar");

}

int delay1()

{

//unsigned int long k;

int i,j;

unsigned int count=0;

for(i=0;i<1000;i++)

{

for(j=0;j<1000;j++)

{

if(!ir)

{

count++;

while(!ir);

}

}

}

return count;

}

void main()

{

welcome();

while(1)

{

int time=delay1();

int RPM=(time\*12)/3;

LCD\_Command(0x01);

LCD\_Command(0x0c);

LCD\_String("RPM");

LCD\_Command(0xc0);

lcd\_data(RPM);

}

}

# APPENDIX - III



16X2 Display Datasheet

