**Objective:**

A tachometer is a device used to measure the **RPM**or Revolutions Per Minute of any rotating body. The word tachometer is derived from two Greek words, tachos means “speed” and metron means “to measure”. It works on the principle of a tachometer generator, which means when a motor is operated as a generator, it produces the voltage according to the velocity of the shaft. It is also known as revolution-counter, and its operating principle can be electromagnetic, electronic or optical-based. Tachometers can be contact based or non-contact ones. Our topic is non contact tachometer. A tachometer that does not need any physical contact with the rotating shaft is called as noncontact digital tachometer. In this type, a laser or an optical disk is attached to the rotating shaft, and it can be read by an IR beam or laser, which is directed by the tachometer.

**List of components:**

1. **Arduino nano**
2. **IR sensor module**
3. **LCD**
4. **100kΩ potentiometer**
5. **Connecting wires**
6. **Bread board**

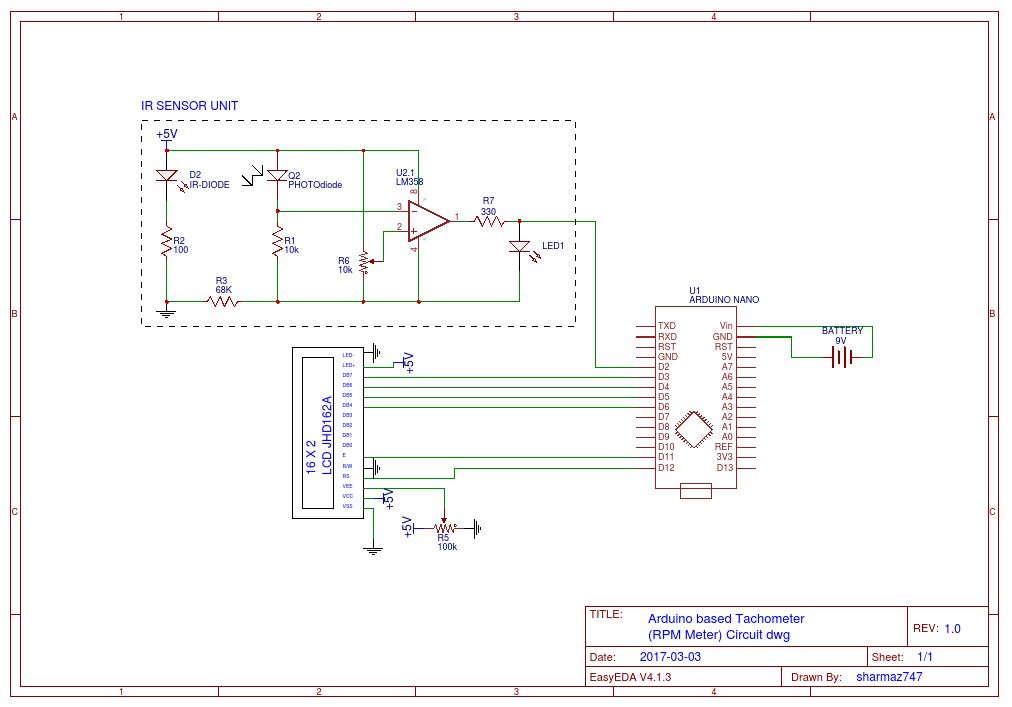
BASIC PRINCIPLE :

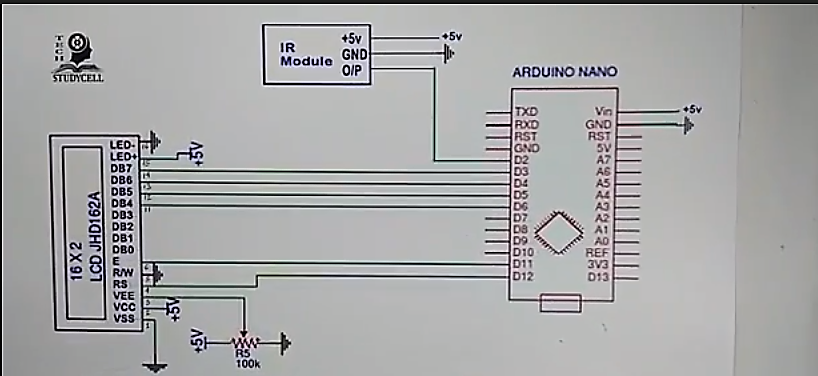
The basic principle involves a simple embedded system with a sensor a controller and LCD. The sensor used here is the one consisting of a white LED and a phototransistor, the controller used is the microcontroller loaded with a compiled code and the actuator is a display device, displaying the speed of the motor. The sensor senses the speed of the motor without actually being in contact with it by the principle of white 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.

When a shaft/body rotates, same point on it repetatively comes to original position after one revolution. But for that it requires some time. By measuring time (in second) between these two positions ie. completeing one revolution, we get time (second) required to complete one revolution. Simply taking reciprocal of this value we get rps (revolution per second) of motor. Multiplying it by 60 will give rpm.

.

Circuit diagram:





As shown in the above **tachometer circuit,** it contains Arduino nano, IR sensor module, and LCD.. IR sensor module output pin is directly connected to pin D2. Vcc and GND are connected to 5V and GND.A 16x2 LCD is connected with arduino nano. Control pin RS, RW and En are directly connected to arduino pin 12, GND and 11. And data pin D4-D7 is connected to digital pins 6, 5, 4 and 3 of arduino.

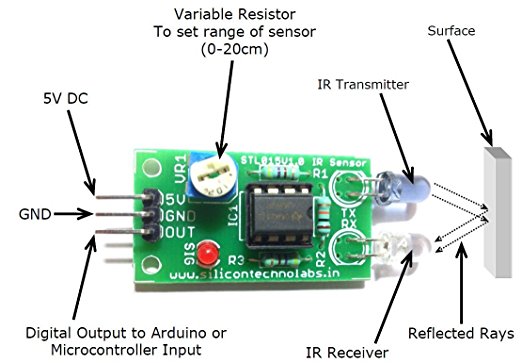
**Working Principle of RPM counter:**

When the motor starts rotating the IR module generate a pulse for each full revolution and this pulse fed to Arduino D2 pin. After receiving the pulse from the IR module Arduino count the total number of pulse receive in 1 second and calculate the speed of the motor in RPM scale with the help of installed program. Then the value or reading is transmitted through 4-bit parallel data pin to LCD display. Thus we can see the motor speed [RPM] in LCD display in real time

**Sensor:**

The sensor circuit consists of an IR transmitter and an IR receiver. An IR LED is used as the transmitter and a photo diode is used as the receiver. A reflective type of IR sensor is used in this project. In this type, the IR transmitter and receiver are placed side by side.

When the IR sensor is powered, the IR transmitter starts emitting IR rays. A motor is placed in front of the IR sensor, with its shafted marked with a white dot. As the motor shaft rotates such that the white spots comes in contact with the sensor, the IR rays are reflected by the dot and falls on the IR receiver. The photo diode, which is used as the IR receiver, starts conducting whenever the IR rays are reflected. At this point, the output of the IR sensor is given to the comparator and the output of the comparator is HIGH when the IR rays are reflected and the output of the comparator is LOW when there are no reflections. Hence, the output of the comparator is in the form of an ON-OFF pulse. This pulse is given to the microcontroller as a timer input and the microcontroller is programmed to calculate the number of times the motor rotates in a second. The speed of the motor is calculated by multiplying the value of final count by 60 to get the speed in revolutions per minute. This value is then displayed on the LCD display.

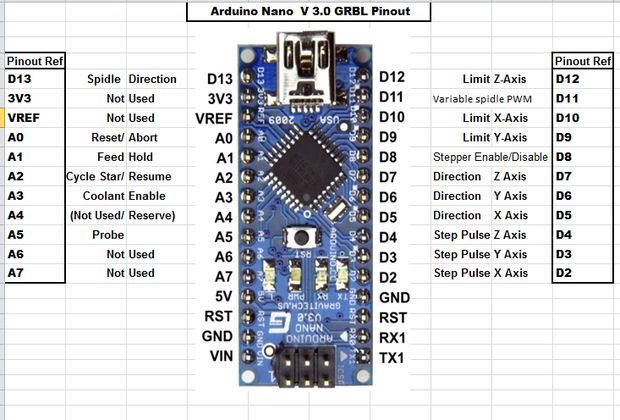


**Arduino nano:**

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328P; offers the same connectivity and specs of the UNO board in a smaller form factor. This controller used is the microcontroller loaded with a compiled code and the actuator is a display device, displaying the speed of the motor. This output pin is D3 D4 D5 D6.

# **Arduino Nano Specifications:**

|  |  |
| --- | --- |
| Microcontroller | Atmel ATmega168 or ATmega328 |
| Operating Voltage (logic level) | 5 V |
| Input Voltage (recommended) | 7-12 V |
| Input Voltage (limits) | 6-20 V |
| Digital I/O Pins | 14 (of which 6 provide PWM output) |
| Analog Input Pins | 8 |
| DC Current per I/O Pin | 40 mA |
| Flash Memory | 16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader |
| SRAM | 1 KB (ATmega168) or 2 KB (ATmega328) |
| EEPROM | 512 bytes (ATmega168) or 1 KB (ATmega328) |
| Clock Speed | 16 MHz |
| Dimensions | 0.73″ x 1.70″ |
| Length | 45 mm |
| Width | 18 mm |
| Weigth | 5 g |



**LCD:**

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. The reasons being LCD are economical, easily programmable have no limitation of displaying . A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers namely Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD.

### Pin Description:

|  |  |  |
| --- | --- | --- |
| **Pin No** | **Function** | **Name** |
| 1 | Ground (0V) | Ground |
| 2 | Supply voltage; 5V | Vcc |
| 3 | Contrast adjustment; through a variable resistor | VEE |
| 4 | Selects command register when low; and data register when high | Register Select |
| 5 | Low to write to the register; High to read from the register | Read/write |
| 6 | Sends data to data pins when a high to low pulse is given | Enable |
| 7 | 8-bit data pins | DB0 |
| 8 | DB1 |
| 9 | DB2 |
| 10 | DB3 |
| 11 | DB4 |
| 12 | DB5 |
| 13 | DB6 |
| 14 | DB7 |
| 15 | Backlight VCC (5V) | Led+ |
| 16 | Backlight Ground (0V) | Led- |

CODE:

#include<LiquidCrystal.h>

LiquidCrystal lcd(12,11,6,5,4,3);

float value=0;

float rev=0;

int rpm;

int oldtime=0;

int time;

void isr() //interrupt service routine

{

rev++;

}

void setup()

{

lcd.begin(16,2); //initialize LCD Display

attachInterrupt(0,isr,RISING); //attaching the interrupt

}

void loop()

{

delay(1000);

detachInterrupt(0); //detaches the interrupt

time=millis()-oldtime; //finds the time

rpm=(rev/time)\*60000; //calculates rpm

oldtime=millis(); //saves the current time

rev=0;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("\_\_\_ TACHOMETER \_\_\_");

lcd.setCursor(0,1);

lcd.print( rpm);

lcd.print(" RPM");

lcd.print(" ");

attachInterrupt(0,isr,RISING);

}

**Explanation and Calculation:**

This program basically monitors the IR sensor's value constantly and with the highest priority using Interrupts*.* Calculate theactual RPM*,* we need the time taken for one revolution*.* And the time taken for one full revolutions is (millis () - time**)**

In this case, let **t**be the time taken for one full revolution , so the total number of revolutions RPMin 60sec( 60\*1000 millisecond ) isrpm = (actual REV/t) \*60000=> rpm = (rev/time)\*60000;

**Performance:**

It can measure RPM over 20k. The performance of this project is not so high. We use a USB fan for this project. Ideally this fan rpm is 4200. In our project we calculated the rpm of this fun is 7800…………………………..

The cost of this project is 1500tk. The components of this project are available in market.

**Applications**

1. The Contactless Digital Tachometer circuit can be used to calculate speed of rotating wheels, discs and motor shafts.

2. This circuit can be used at homes to check speed of small battery operated fans and other motor based devices

**Limitations of the project:**

1. Sensor range extends up to 2~6 cm. Over this range rpm does not remains stable.
2. PCB board is more preferable than bread board. Beard board has loss connection.
3. We use USB fan for this project. The good way is use laptop or pc cooling fan to gives a good accuracy.

**Conclusion:**

Although this project accuracy is not good but it is good project. Just need to few components to build this project. This is very simple project and very easy to measure rpm . Though there are many tachometers available in the market, this device is comparatively cheap and works quite well.

**Reference:**

1. https://www.electronicshub.org/contactless-digital-tachometer-using-8051-microcontroller

2. http://www.instructables.com/id/Simple-Motor-Speed-Tester-Tachometer

3.http://electricdiylab.com/how-to-make-arduino-based-digital-tachometer-simple-diy-tutorial/

4. http://www.gravitech.us/arna30wiatp.html