VISVESVARAYA NATIONAL INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

ECP320 : LINEAR ELECTRONIC CIRCUITS LAB

PROJECT SYNOPSIS

NON CONTACT TACOMETER

BY:

BT23EEE101

ARYAN

BT23EEE098

MINHAJ

BT23EEE097

SANKET

Digital Tachometer Using IR Sensor, Op-Amp, Transistor, and Arduino

1. Objective

The objective of this project is to design and implement a digital tachometer capable of accurately measuring the rotational speed (RPM) of a rotating object using a non-contact infrared sensor system. The setup uses an IR LED and phototransistor to detect motion, an Op-Amp comparator and transistor for signal conditioning, and an Arduino microcontroller for frequency measurement and RPM calculation.

2. Introduction

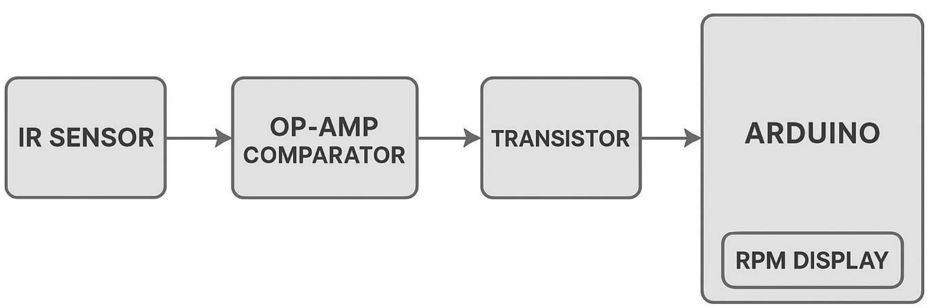
Tachometers are essential instruments used to measure the rotational speed of mechanical systems. Traditional contact-based tachometers suffer from wear and tear and are not suitable for high-speed or sensitive applications. A digital tachometer, on the other hand, provides a non-contact, efficient, and reliable method of measuring RPM, making it ideal for modern engineering systems. This project aims to create a cost-effective and precise tachometer using commonly available electronic components.

3. Principle of Operation

The tachometer functions by detecting light reflected from a rotating object marked with reflective tape. The IR LED continuously emits infrared light, which is reflected back when a reflective surface passes in front of it. The phototransistor detects this reflected light, producing an analog signal corresponding to the light intensity.

This signal is passed through an Op-Amp comparator circuit, which transforms the analog waveform into a digital square wave. A transistor stage is used to strengthen the signal before it is fed into an Arduino. The Arduino counts the number of pulses received within a predefined time interval and calculates the RPM accordingly.

4. Block Diagram

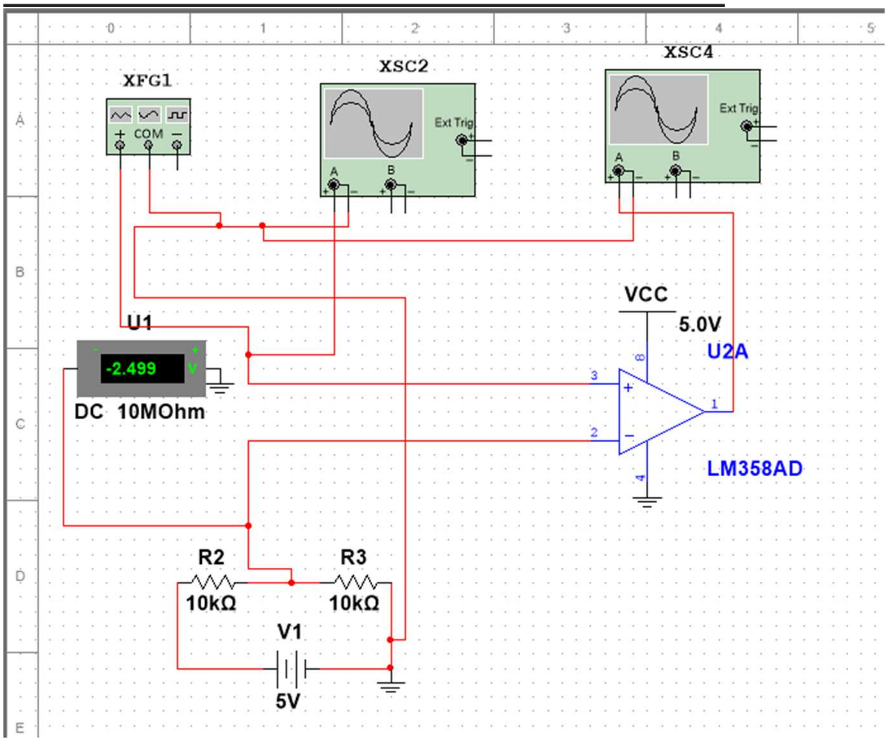


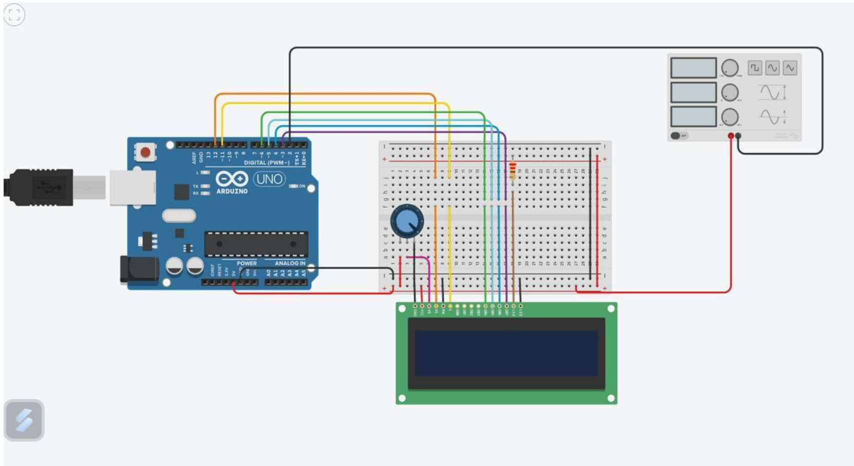
5. Components Required

- IR LED (1)
- IR Phototransistor / IR Sensor Module (1)
- IC741 / LM358 Op-Amp (1)
- Arduino Uno/Nano (1)
- 16x2 LCD Display with I2C module (optional)
- Resistors (10kΩ)
- Breadboard and Jumper wires
- Power Supply (5V)
- Reflective tape or black-white marker

6. Circuit Description

The function generator XFG1 here is the analogy of the IR sensor which is deducing the rotations in a wave form





• The IR LED is forward biased and emits infrared light continuously. It is aimed at a rotating shaft with reflective tape.
• The phototransistor is arranged to receive the reflected IR light. When the reflection is detected, the phototransistor conducts, producing a varying voltage signal.
• The Op-Amp is configured as a comparator. It compares the phototransistor output with a reference voltage. When the input crosses the threshold, the output of the Op-Amp toggles between high and low, producing a square wave.
• The transistor is used to buffer and amplify this signal before feeding it into the Arduino.
• Arduino is programmed to count rising or falling edges in the signal for a 1-second interval and calculate RPM.

7. Working and Calculations

The Arduino counts the number of pulses (i.e., revolutions) detected in a fixed time (typically 1 second). The formula used to calculate RPM is:

RPM = (Number of Pulses per Second) × 60

If there are multiple reflective markers per revolution, the count is divided

accordingly: RPM = (Pulse Count × 60) / Number of Pulses per Revolution

8. Arduino Implementation

• Uses interrupt or digital pin polling to detect signal edges
• Performs real-time RPM calculations
• Displays results on the Serial Monitor or LCD
• Provides fast, accurate, and programmable measurement
• CODE:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(8, 9, 13, 12, 11, 10);

volatile int rev = 0;
volatile unsigned long lastPulseTime = 0;
```

```
unsigned long lastMillis = 0;
int rpm = 0;

void isr() {
  unsigned long now = millis();  // Check in milliseconds

  // Accept only 1 pulse every 50 ms (adjust if needed)
  if (now - lastPulseTime > 50) {
    rev++;
    lastPulseTime = now;
  }
}

void setup() {
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("TACHOMETER");

  pinMode(2, INPUT);
  attachInterrupt(digitalPinToInterrupt(2), isr, RISING);

  Serial.begin(9600);
}

void loop() {
  unsigned long currentMillis = millis();

  if (currentMillis - lastMillis >= 1000) {
    noInterrupts();
    int count = rev;
    rev = 0;
    interrupts();

    rpm = count * 60;

    lcd.setCursor(0, 1);
    lcd.print("RPM: ");
    lcd.print(rpm);
    lcd.print("    ");  // Clear trailing digits

    Serial.print("Filtered Pulses: ");
    Serial.print(count);
    Serial.print(" | RPM: ");
    Serial.println(rpm);

    lastMillis = currentMillis;
  }
    }
```

9. Applications

- Motor speed monitoring
- Rotating machinery diagnostics
- Industrial automation systems
- Robotics and mechatronics projects
- Vehicle speedometers (non-contact type)

10. Conclusion

This project will successfully demonstrates a cost-effective and reliable method to measure RPM using an IR sensor system, Op-Amp comparator, transistor amplifier, and Arduino microcontroller. It is suitable for academic and industrial applications where non-contact, real-time speed monitoring is required.