Report

EDC Lab Project

Submitted to: DR. SURENDER HANS

Group members:

- Rounak kumar (2023UEE2021)
- Pragati Varshney (2023UEE2020)
- Baibhav kumar (2023UEE2023)

IR Tachometer Using Arduino Nano, OLED Display, and 9V Battery Power Supply

Objective

The goal of this project is to design and build a portable IR Tachometer using an **Arduino Nano**, an **OLED display**, and a **9V battery** as the power source. The tachometer will measure the rotational speed (RPM) of a rotating object and display the results in real-time on the OLED screen.

Components Required

1. **Arduino Nano** – Microcontroller to process the sensor data and calculate RPM.
2. **IR Sensor Module** – Detects reflections from a rotating object and sends pulses to Arduino.

3. **OLED Display (128x64)** – Displays the calculated RPM value.
4. **9V Battery** – Powers the Arduino Nano and peripherals.
5. **Jumper Wires** – To connect the components.
6. **Reflective Marker** – A small reflective sticker or tape placed on the rotating object.
7. **9V Battery Connector Clip** – To connect the 9V battery to the Arduino Nano.

Working Principle

1. **IR Sensor Detection**:

The IR sensor emits infrared light, and when it detects the reflection from a reflective surface on the rotating object, it generates a pulse. The Arduino Nano counts these pulses using an interrupt service routine.

2. **RPM Calculation**:

The RPM is calculated based on the number of pulses detected in one second, using the formula:

$RPM = Number\ of\ Pulses\ in\ 1\ Second \times 60 / Reflective\ Markers$

In this case, each reflective marker produces one pulse per rotation.

3. **Display on OLED**:

The calculated RPM value is shown on the OLED display in real-time.

Circuit Design

1. **Power Supply (9V Battery)**:
    a. Connect the **positive terminal** of the 9V battery clip to the **VIN** pin on the Arduino Nano.
    b. Connect the **negative terminal** of the 9V battery clip to the **GND** pin on the Arduino Nano.
2. **IR Sensor Connections**:
    a. **VCC** → 5V pin on Arduino Nano.
    b. **GND** → GND pin on Arduino Nano.
    c. **OUT** → D2 pin on Arduino Nano (for pulse detection).
3. **OLED Display Connections**:
    a. **VCC** → 3.3V pin on Arduino Nano.
    b. **GND** → GND pin on Arduino Nano.
    c. **SCL** → A5 (I2C Clock).
    d. **SDA** → A4 (I2C Data).

Arduino Code

The Arduino code counts the pulses from the IR sensor, calculates the RPM, and displays the result on the OLED display. Below is the code used for this project:

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// Define OLED display width and height
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define SCREEN_ADDRESS 0x3C

// Initialize the OLED display
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// Define the sensor pin
#define SENSOR_PIN 2

volatile unsigned int spokeCount = 0; // Count the number of spokes passed
unsigned long lastTime = 0; // Track the last time RPM was calculated
unsigned long lastUpd = 0; // Track the last time RPM was calculated
const unsigned long interval = 500; // Update interval in milliseconds
const unsigned long updInterval = 3;


// Spoke count for a full wheel revolution
const int SPOKES = 3;

void setup() {
  // Initialize serial communication
  // Serial.begin(9600);

  // Initialize the sensor pin
  pinMode(SENSOR_PIN, INPUT);
  attachInterrupt(digitalPinToInterrupt(SENSOR_PIN), countSpoke, FALLING);

  // Initialize the OLED display
  if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
    Serial.println("SSD1306 allocation failed");
```

```
    for (;;);
  }

  display.clearDisplay();
  display.setTextSize(1); // Normal 1:1 pixel scale
  display.setTextColor(SSD1306_WHITE); // Draw white text
  display.setCursor(0, 0);
  display.println("Measuring RPM...");
  display.display();
}

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

  // Calculate RPM every interval

  if (currentTime - lastTime >= interval ) {
    detachInterrupt(digitalPinToInterrupt(SENSOR_PIN)); // Prevent interrupt during
calculation
    unsigned int count = spokeCount;
    attachInterrupt(digitalPinToInterrupt(SENSOR_PIN), countSpoke, FALLING);

    // Calculate RPM
    float rpm = (count / (float)SPOKES) * (60000.0 / interval);

    // Display RPM on OLED
    display.clearDisplay();
    display.setCursor(0, 0);
    display.println("Wheel RPM:");
    display.setTextSize(2);
    display.setCursor(0, 20);
    display.print(rpm, 1); // Display with 1 decimal precision
    // display.setCursor(0,40);
    // display.println("Spokes:");
    // display.print(spokeCount,1);
    display.display();

    // Serial.print("RPM: ");
    // Serial.println(rpm);

    spokeCount = 0; // Reset spoke count
    lastTime = currentTime;
  }
```

```
}

// Interrupt Service Routine to count spokes
void countSpoke() {
  unsigned long currentTime = millis();
  if(currentTime - lastUpd >= updInterval)
  {
    // Serial.println(spokeCount);
  spokeCount++;
  lastUpd = currentTime;
  }

}
```

Results and Observations

1. **Power Efficiency**:
    a. The 9V battery provides a practical power source for this portable device. The Arduino Nano's efficient power consumption ensures reasonable runtime with the 9V battery.
2. **Portability**:
    a. The 9V battery makes the tachometer easy to use in field applications where a direct power source may not be available.
3. **Accuracy**:
    a. The tachometer accurately measures RPM and displays the result on the OLED screen in real-time.
4. **Battery Life**:
    a. The 9V battery offers about **7 hours** of continuous operation, depending on the current draw of the components (Arduino Nano, OLED, and IR sensor).

Applications

1. **Motor Speed Monitoring**:
    a. Used to monitor the speed of motors, turbines, or fans in industrial settings.
2. **Portable RPM Measurement**:
    a. Ideal for field testing where mobility is required.

3. **Educational Projects**:
    a. A simple and effective project for learning about sensors, microcontrollers, and display systems.

Conclusion

This project successfully demonstrates the use of an IR sensor, Arduino Nano, and an OLED display to measure and display the RPM of a rotating object. The 9V battery provides a convenient power source for portable use, making the tachometer ideal for mobile applications. The device is compact, accurate, and efficient, offering a valuable tool for measuring rotational speed in various fields.

Future Enhancements

1. **Longer Battery Life**:
    a. Use a **Li-ion battery** or **AA battery pack** for longer usage time and rechargeable capabilities.
2. **Wireless Monitoring**:
    a. Integrate **Bluetooth** or **Wi-Fi** to send RPM data to a smartphone or computer for remote monitoring.
3. **Data Logging**:
    a. Add an **SD card module** to store RPM readings for analysis over time.
4. **Multi-Functionality**:
    a. Expand the functionality of the device to measure other parameters such as temperature or voltage.