# Digital Speedometer using Arduino Board for Formula Student Cars

Kshitij Kolhe
Second Year Mechanical Department
JSPM's Rajarshi Shahu College of Engineering
Pune, India

Omkar Taru
Third Year Mechanical Department
JSPM's Rajarshi Shahu College of Engineering
Pune, India

***Abstract*— This paper outlines the design and implementation of Digital speedometer system using Arduino. The Arduino board senses the signals receiving from sensor as an input then do some Arithmetic calculations and showed the results on Liquid crystal display (LCD) as an output in kmph.**

**This article uses a GS101201 speed sensor to measure the speed of one of the cars wheels. The Arduino board receives signals from the sensor and calculates the rpm then converted into a speed, and send this information out to the LCD screen on the dashboard of the car. It is compatible with any kind of wheel, simply changing the radius of the wheel in the program in Arduino IDE software. Particularly it is very useful to monitor the speed of Formula student cars, motors, turbines using this Arduino system.**

***Keywords*— Arduino Nano, Adruino IDE, speed sensor, microcontroller, hardware, software, LCD, buck converter, tachometer, etc.***

## I. INTRODUCTION

Pretty much all speedometers produced until the 1980s worked using the eddy current and cable mechanism much like Schulz's original, patented design. But there are drawbacks. First, there are lots of mechanical parts to wear out (which makes them inaccurate) or fail suddenly. If the speedometer cable breaks, the whole contraption instantly becomes useless and it takes a mechanic to make a repair. Long speedometer cables are particularly impractical, which has always been something of a problem in large commercial vehicles such as trucks and buses And it's not just the cable that poses a problem, it can be difficult to read a conventional speedometer dial if you're racing down the freeway, especially at night. Do you really want to take your eyes off the road to figure out where the needle is on the dial? Some people prefer to see their speed as a simple number on a well-lit digital display.

Electronic devices can be analog or digital. Values on analog devices are (normally) infinitely variable. A speedometer that shows a car's speed by means of a dial is an analog device. The hand on that dial moves smoothly around the dial and can take any value that the car's engine can create. In a digital device, values are represented by numbers and therefore do not have the variability of analog devices.

## II. BLOCK DIAGRAM OF THE SYSTEM

The system consists of hardware and software part. This fig. Shows the block diagram of the Digital speedometer. As for the software part plays biggest role in system. All the program is stored in Arduino nano using c language.

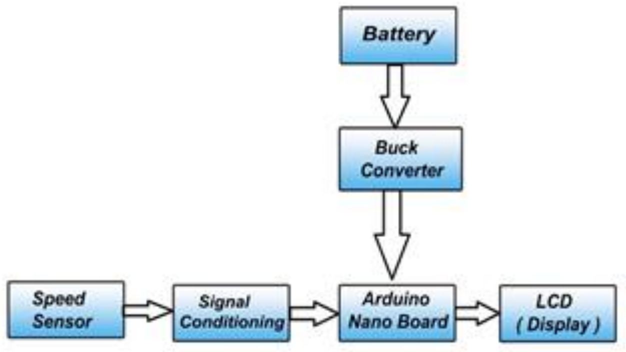


Fig. 1. Block diagram of the system.

As shown in fig Arduino nano is used as a main processing device that input is connected as to give some amplified signals to the Arduino board and it process and do some calculations on that signals and showed accurate output on the LCD (display).

## III. HARDWARE SYSTEM

There are some important components that used for building Digital speedometer which is given below.

### *A. Arduino Nano*

Arduino Nano is a small, compatible, flexible and breadboard friendly Microcontroller board, developed by Arduino.cc in Italy, based on ATmega328p.

Arduino Nano Pinout contains 14 digital pins, 8 analog Pins, 2 Reset Pins & 6 Power Pins. Each of these Digital & Analog Pins are assigned with multiple functions but their main function is to be configured as input or output.

Arduino Nano comes with a crystal oscillator of frequency 16 MHz. It is used to produce a clock of precise frequency using constant voltage. There is one limitation using Arduino Nano it doesn't come with DC power jack, means you cannot supply external power source through a battery, therefore we used buck converter in circuit.

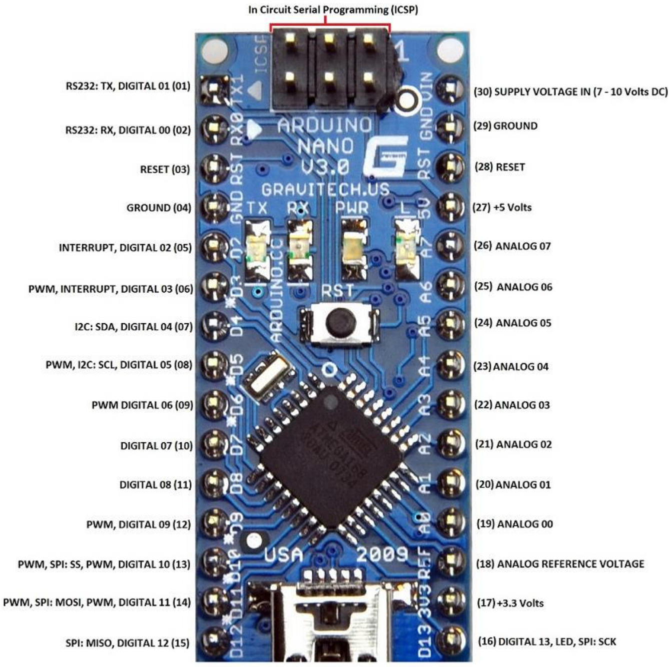
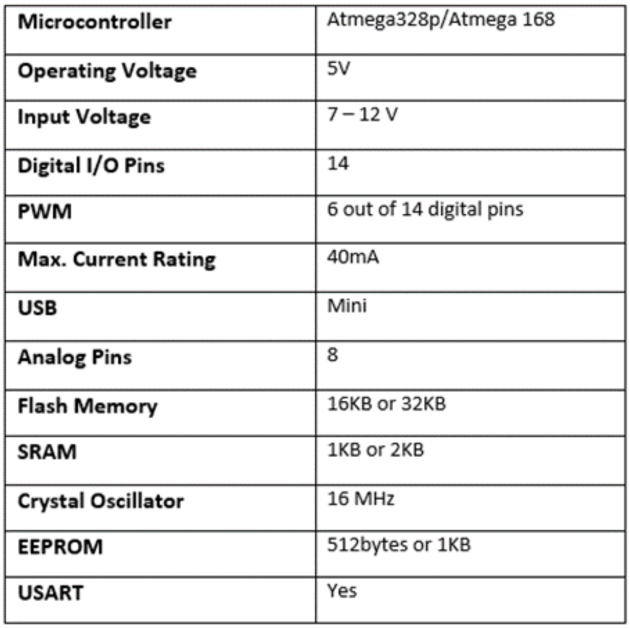

In Circuit Serial Programming (ICSP)
RS232: TX, DIGITAL 01 (01)
RS232: RX, DIGITAL 00 (02)
RESET (03)
GROUND (04)
INTERRUPT, DIGITAL 02 (05)
PWM, INTERRUPT, DIGITAL 03 (06)
I2C: SDA, DIGITAL 04 (07)
PWM, I2C: SCL, DIGITAL 05 (08)
PWM DIGITAL 06 (09)
DIGITAL 07 (10)
DIGITAL 08 (11)
PWM, DIGITAL 09 (12)
PWM, SPI: SS, PWM, DIGITAL 10 (13)
PWM, SPI: MOSI, PWM, DIGITAL 11 (14)
SPI: MISO, DIGITAL 12 (15)
(30) SUPPLY VOLTAGE IN (7 - 10 Volts DC)
(29) GROUND
(28) RESET
(27) +5 Volts
(26) ANALOG 07
(25) ANALOG 06
(24) ANALOG 05
(23) ANALOG 04
(22) ANALOG 03
(21) ANALOG 02
(20) ANALOG 01
(19) ANALOG 00
(18) ANALOG REFERENCE VOLTAGE
(17) +3.3 Volts
(16) DIGITAL 13, LED, SPI: SCK
ARDUINO NANO V3.0
GRAVITECH.US
USA 2009

Fig. 2. Arduino Nano.

Flash memory is 16KB or 32KB that all depends on the Atmega board Atmega168 comes with 16KB of flash memory while Atmega328 comes with a flash memory of 32KB. Flash memory is used for storing code. The 2KB of memory out of total flash memory is used for a bootloader.

The SRAM can vary from 1KB or 2KB and EEPROM is 512 bytes or 1KB for Atmega168 and Atmega328 respectively. It is programmed using Arduino IDE which is an Integrated Development Environment that runs both offline and online.

TABLE I. ARDUINO NANO SPECIFICATIONS



| | |
|---|---|
| **Microcontroller** | Atmega328p/Atmega 168 |
| **Operating Voltage** | 5V |
| **Input Voltage** | 7 – 12 V |
| **Digital I/O Pins** | 14 |
| **PWM** | 6 out of 14 digital pins |
| **Max. Current Rating** | 40mA |
| **USB** | Mini |
| **Analog Pins** | 8 |
| **Flash Memory** | 16KB or 32KB |
| **SRAM** | 1KB or 2KB |
| **Crystal Oscillator** | 16 MHz |
| **EEPROM** | 512bytes or 1KB |
| **USART** | Yes |

### B. Gear Tooth speed sensor

Fig. 3. GS101201 gear tooth sensor.

The GS101201 sensor was chosen as the sensor to sense the speed of vehicle by measuring the speed of front wheels because of its high switching frequency.

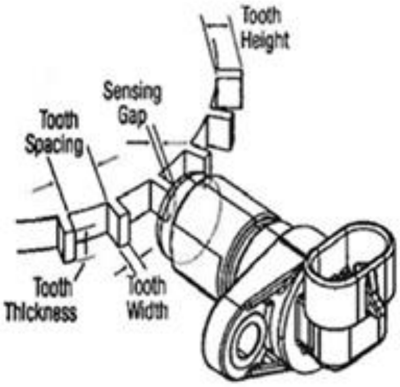


Fig. 4. Sensor positioning.

The sensor is paired with a gear of suitable dimensions as per the datasheet of the sensor. And is assembled as depicted in the figure. (sensing gap is 2-3 mm).

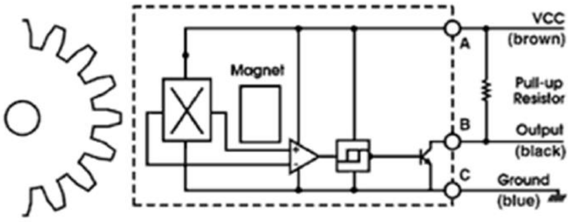


Fig. 5. Sensor construction.

Above figure shows the construction of sensor. The gear is to be made up of a ferromagnetic material for the sensor to work. The sensor is capable of detecting the presence of a ferromagnetic material within its sensing distance. This property is used to measure the wheel speed with the help of a gear.

### C. 16×2 LCD

An LCD is an electronic display module which uses liquid crystal to produce a visible image. 16×2 LCD is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc.

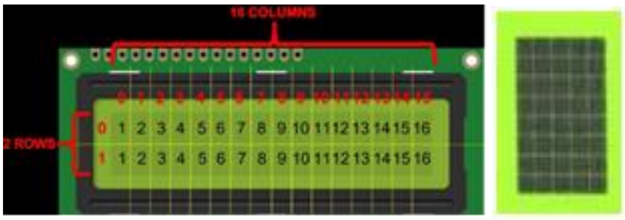


Fig. 6. 16×2 LCD.

It will have (16×2=32) 32 characters in total and each character will be made of 5×8 Pixel Dots. A single character with all its Pixels is shown in the above picture. The advantage of an I2C LCD is that the wiring is very simple. You only need two data pins to control the LCD.

### D. I2C module

I2C is a serial communication protocol so you can connect multiple slaves to a single master (like SPI) and you can have multiple masters controlling single, or multiple slaves.

Fig. 7. I2C madule.

I2C only uses two wires to transmit data between devices as follows:

*1) SDA (Serial Data):* The line for the master and slave to send and receive data.

*2) SCL (Serial Clock):* The line that carries the clock signal A5 pin.

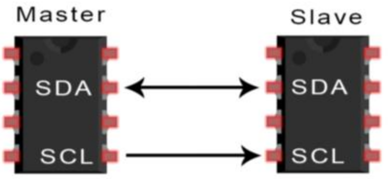


Fig. 8. I2C connection.

so, data is transferred bit by bit along a single wire (the SDA line) A4 pin.

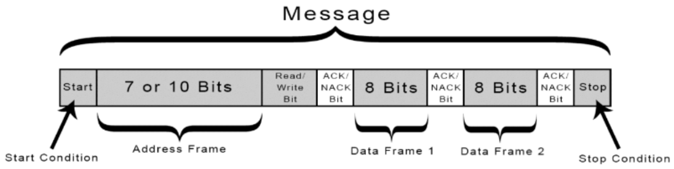


Fig. 9. I2C communication.

With I2C, data is transferred in messages. Messages are broken up into frames of data. Each message has an address frame that contains the binary address of the slave, and one or more data frames that contain the data being transmitted. The message also includes start and stop conditions, read/write bits, and ACK/NACK bits between each data frame.

### E. Buck converter (Step down converter)

A buck converter (step-down converter) is a DC-to-DC power converter which steps down the voltage (while stepping up current) from its input supply to its output load.

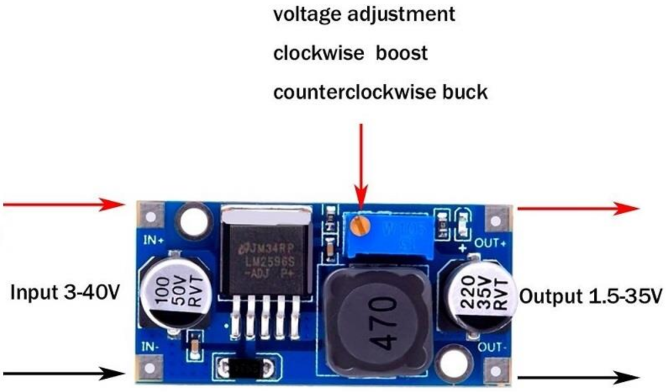


Fig. 10. Buck converter.

In this DC-DC Buck Converter Step Down Module LM2596 Power Supply is a step-down(buck) switching regulator, capable of driving a 3-A load with excellent line and load regulation. The LM2596 series operates at a switching frequency of 150kHz, thus allowing smaller sized filter components than what would be required with lower frequency switching regulators.

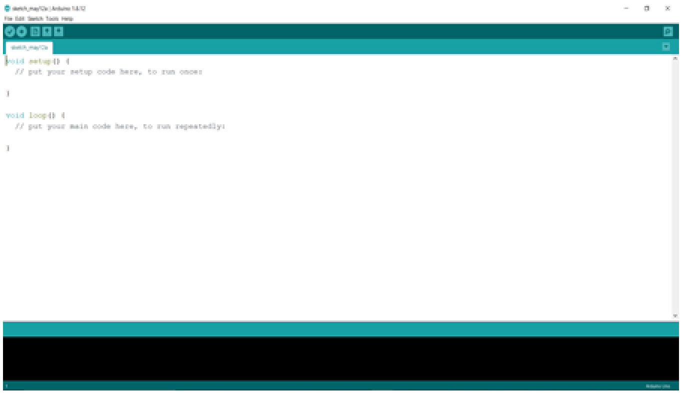
## IV. SOFTWARE SYSTEM

The software that needs to be installed for programming the board and some important concept of programming is provided in this section.

### A. Arduino IDE

Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.

It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Fig. 11. Arduino IDE interface.

A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more. Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.

The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board. The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module. This environment supports both C and C++ languages.

### *B. Interrupts*

Interrupts stop the current work of Arduino such that some other work can be done. The main program is running and performing some function in a circuit. However, when an interrupt occurs the main program halts while another routine is carried out. When this routine finishes, the processor goes back to the main routine again.

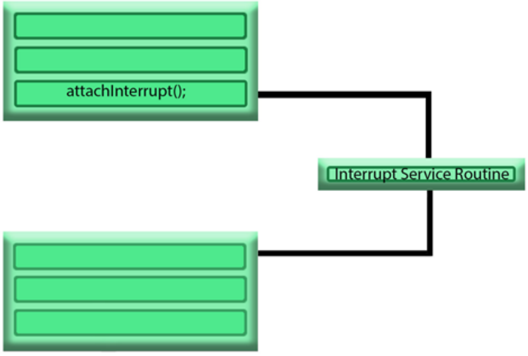


Fig. 12. Interrupt Service Routine, ISR.

Here are some important features about interrupts:

- Interrupts can come from various sources. In this case, we are using a hardware interrupt that is triggered by a state change on one of the digital pins.
- Most Arduino designs have two hardware interrupts (referred to as "interrupt0" and "interrupt1") hard-wired to digital I/O pins 2 and 3, respectively.
- The Arduino Mega has six hardware interrupts including the additional interrupts ("interrupt2" through "interrupt5") on pins 21, 20, 19, and 18.
- You can define a routine using a special function called as "Interrupt Service Routine" (usually known as ISR).
- You can define the routine and specify conditions at the rising edge, falling edge or both. At these specific conditions, the interrupt would be serviced.
- It is possible to have that function executed automatically, each time an event happens on an input pin.

There are two types of interrupts:

1. Hardware Interrupts – They occur in response to an external event, such as an external interrupt pin going high or low.
2. Software Interrupts – They occur in response to an instruction sent in software. The only type of interrupt that the "Arduino language" supports is the attachInterrupt() function.

## V. PROPOSED SYSTEM

This part introduces our approach towards making of digital speedometer.

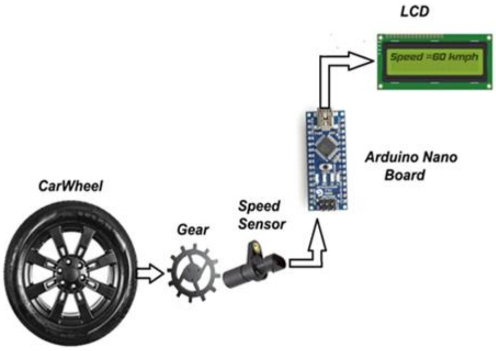


Fig. 13. Flow chart of system.

The sensor used in this setup is capable of detecting the presence of metal. This property can help in many applications, one of them being measuring speed with the help of a microcontroller analysing the data sent by the sensor to measure speed of shafts, in our case a hub. A gear of metallic material of compliant dimensions as per the datasheet of GS101201 sensor is to be made. A concept of such gear is depicted in the CAD figure 14.

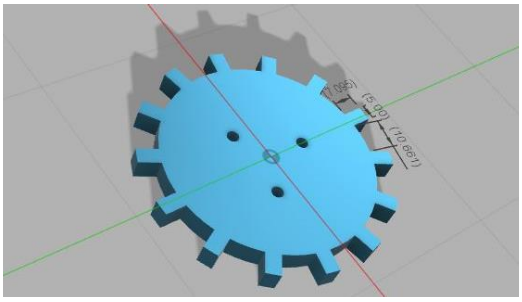


Fig. 14. Gear CAD.

The sensor is to be connected in a way such that when the gear rotates along with the wheel, the teeth point towards the flat surface of the sensor which detects the presence of metal. A concept of such arrangement is depicted in the CAD figure 15.

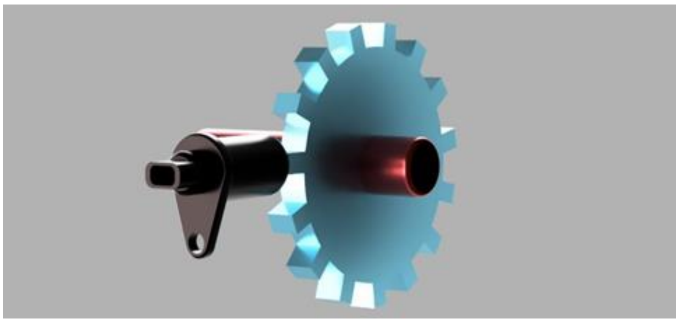


Fig. 15. Sensor and gear arrangement CAD.

The sensor is connected to Arduino Nano board to calculate wheel speed by a logical program saved on the board which is then displayed to the concerned authorities by means of a display (figure 16).

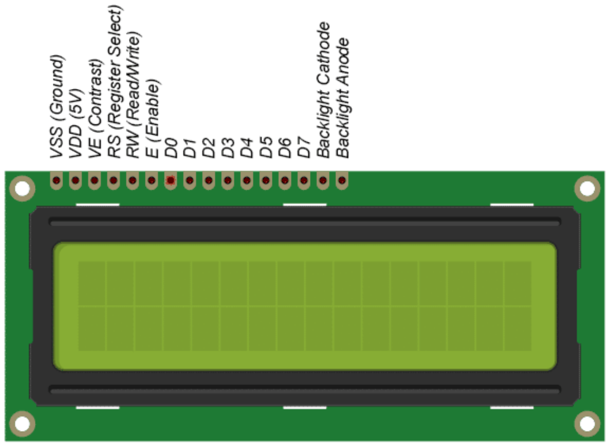


Fig. 16. 16×2 LCD.

## VI. CIRCUIT INTERFACING

The electrical connections to be done are clearly depicted in the figure following.

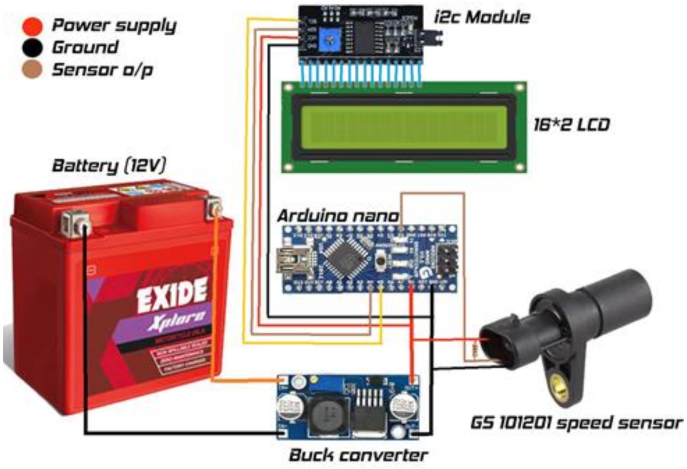


Fig. 17. Wrriring diagram of setup.

The I2C is connected to the display to convert the 8bit data transfer capability of the display into I2C type of data transfer to reduce number of wires in the wiring harness. The 4 terminals of I2C module (5V, GND, SDA, SCL) are connected to the terminals on the Arduino Nano as shown. The sensor works on 5V DC currents hence is powered by connecting to 5V and GND terminals of the Arduino Nano. The output terminal of the sensor is connected to digital pin 2 of the Arduino as the pin supports hardware interrupt necessary for measuring speed with high precision. The Arduino Nano is powered by the 12V battery through a buck converter which supplies 5V constant DC supply free of voltage spikes and drops due to the 12V battery also being used by other components which could be a cause for the same. Connecting wires of suitable gauge and length are used for the connections and are then covered or tied together to avoid tangles.

## VII. ACTUAL SETUP

The following section explains the current setup of the speedometer installed.

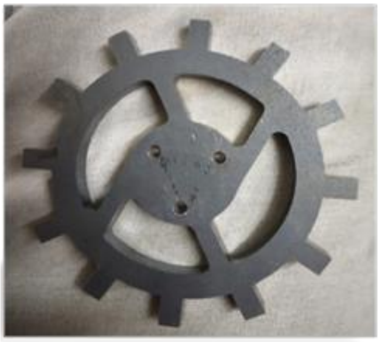


Fig. 18. Mild steel gear.

A mild steel billet is cut into the gear (shown in the figure above) by the process of laser cutting and is mounted on the front right hub (shown in the figure below) with a suitable coupling and hence rotates along with the wheel.

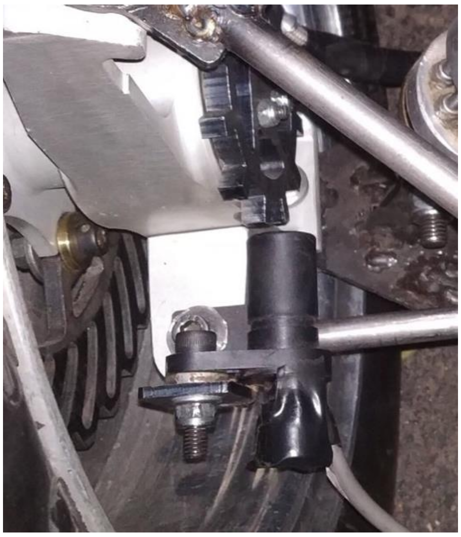
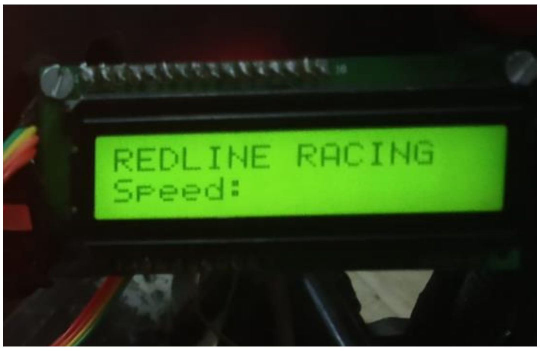


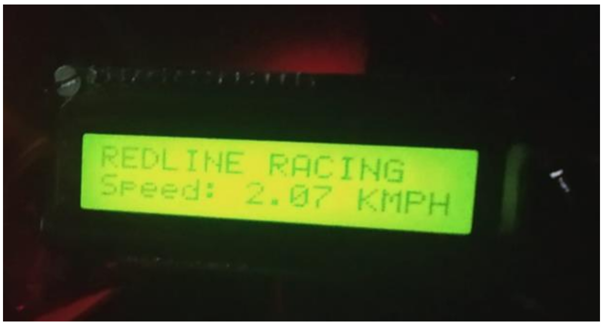
Fig. 19. Actual setup on vehicle.

The calculations are done by Arduino Nano to which the sensor is connected which sends a pulse to the Arduino Nano every time a tooth passes by the sensor. The time difference is calculated between two pulses by the built-in functions of Arduino Nano and with the help of a suitable code, false readings are identified and scrapped to get accurate time difference. The reciprocal of this time difference gives the frequency of the rotating gear. The frequency is then divided by the number of pulses per revolution i.e. the number of teeth on the gear gives number of revolutions per second which the multiplied by 60 gives number of revolutions per minute (RPM). The car wheel is 16 inches in diameter, the RPM is multiplied by a factor of 0.005911 to obtain the speed of vehicle in Kmph units. Arduino Nano's digital pin number 2 is used as the signal input to the board from the sensor as it is one of the pins that supports interrupt service routine that helps to obtain readings with the highest speed possible for the board. The sensor is powered by getting 5-volt constant D.C. supply from Arduino Nano and is also grounded to the board itself. A pull up resistor of 1K Ohm is connected between the VCC and output terminals of the sensor. The Arduino Nano is powered by a buck converter that converts the 12-volt supply from the battery to 5-volt supply suitable for the board. The speed calculated is displayed to the driver through a suitable liquid crystal display in kmph (shown in the figures 20 and 21).



```
REDLINE RACING
Speed:
```

Fig. 20. 16×2 LCD.



```
REDLINE RACING
Speed: 2.07 KMPH
```

Fig. 21. Displaying speed.

## VIII. PROCUREMENT OF MATERIALS

This section includes the necessary materials needed to build the project.

A list of parts is mentioned in the table below that were needed for our setup is mentioned below with their costs and quantity.

TABLE II. COST OF PRODUCTS INVOLVED

| Sr No. | Name of Part | Quantity | Cost |
|---|---|---|---|
| 1 | Arduino Nano | 1pcs | ₹ 220 |
| 2 | GS101201 Sensor | 1pcs | $59.42 |
| 3 | 16×2 LCD | 1pcs | ₹ 125 |
| 4 | I2C Module | 1pcs | ₹ 79 |
| 5 | Buck Converter | 1pcs | ₹ 59 |
| 6 | Arduino IDE | - | NIL |
| 7 | Gear | 1pcs | ₹100 |
| 9 | Rainbow wires | 2 m | ₹60 |
| 10 | M6 Allen Nut, Bolt | 1pair | ₹5 |

The costs mentioned in the above table are subject to change, as per market conditions and specific project requirements.

## IX. OTHER APPLICATIONS

This setup can also be used in many other places where the need of measuring the speed of shafts arises. Some examples are mentioned below but there are more applications of the setup than only the ones mentioned below:

1. Water and steam turbines speed measurement
2. Windmill speed measurement
3. Automobile industries
4. Motor speed measurement, etc.

An example for potential use of this setup is shown below.

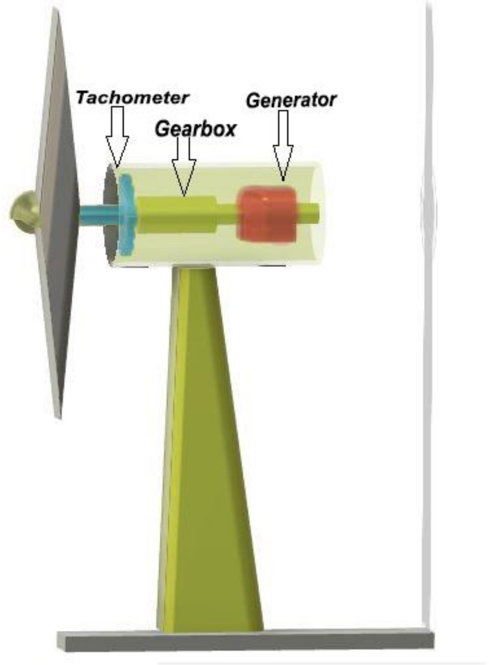


Fig. 22. Potential use.

## X. CONCLUSION

This article proposed a new approach for low cost flexible setup for observing the precise speed of any kind of wheel and also any weather conditions. The results showed that the system works accurate and successfully.

Future research is Arduino based data logger which measures the speed using GPS module with the time stamp for monitoring the behavior of a car and stores that data into the SD card.

## ACKNOWLEDGMENT

The authors would like to thank Team Redline Racing for allowing and helping them to implement and test this setup to measure the speed of the car as explained above.

## REFERENCES

[1] Chris Woodford, 'Speedometers', Jan 2020, [Online]. Available: https://www.explainthatstuff.com/how-speedometer-works.html [Accessed : 30-September -2019 ].

[2] ARDUINO.CC, 'Arduino – Introduction', 2015, [Online]. Available: http://arduino.cc/en/Guide/Introduction. [Accessed : 23-October -2019 ].

[3] Adnan Aqeel, 'Introduction to Arduino Nano', [Online]. Available : https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html [Accessed : 24- October -2019].

[4] Adnan Aqeel 'Introduction to Arduino IDE', [Online]. Available : https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html. [Accessed : 10- November 2019].

[5] Tutorials point, 'Arduino – Interrupts'[Online]. Available : https://www.tutorialspoint.com/arduino/arduino_interrupts.htm [Accessed : 20 November 2019].

[6] 'GS101201 – GS101202 speed sensor'[Online]. Available : https://switches-sensors.zf.com/us/product/speed-sensor-gs1012/ [Accessed : 30- December-2019].

## AUTHORS

Omkar Taru is currently in the third year of Bachelors of Engineering in Mechanical Engineering from the JSMP's Rajarshi Shahu College of Engineering affiliated to Pune University, India. He received his Diploma in Mechanical Engineering from the Navsahyadri Institutes of Technology Pune, India in 2017. His research interests include automation and mechatronic projects using microcontrollers and embedded Systems.

Kshitij Kolhe is currently in the second year of Bachelors of Engineering from the discipline of Mechanical Engineering from JSMP's Rajarshi Shahu College of Engineering affiliated to Pune University, India. His research interests include data acquisition systems and mechatronics using microcontrollers and embedded systems. Completed industrial robotics training course.