BICYCLE SPEEDOMETER AND ODOMETER

INTRODUCTION

Computerized measurements have become an indispensable part of mechanical systems in all its disciplines. With electronic measuring tools, the extent to which systems can be measured elevates in accuracy, precision, and orders of magnitude. As such, the following project involves the design and construction of a digital bicycle speedometer and odometer.

This report details the implementation of Arduino circuitry and Hall effect sensor to measure, compute and display instantaneous cycling speed and cumulative distance travelled.

DEVICE HARDWARE DESIGN & JUSTIFICATION

Arduino MEGA 2560 REV3

Serves as the embedded system platform for hardware and software components. Contains bidirectional I/O digital pins responsible for relaying data from the sensor and outputting processed data by means of the Arduino programming software, Sketch. Furthermore, digital SDA and SCL pins allows for I2C master/slave data transmission and clock signal for communication timing, respectively **(Fig. 1.1)**.

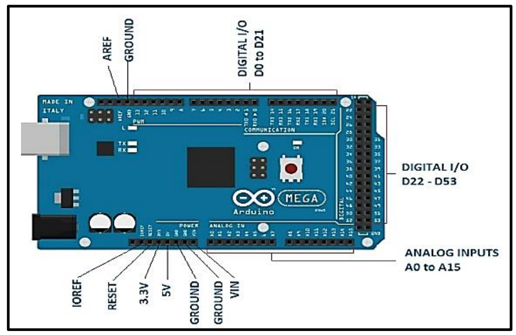

Figure 1.1: Arduino MEGA diagram

Hall Effect Sensor A3144

Unipolar digital switching characteristic appropriates a binary "on" and "off" position, corresponding whether the magnet's magnetic flux exceeds the sensor's pre-established threshold **(Fig 1.2)**.

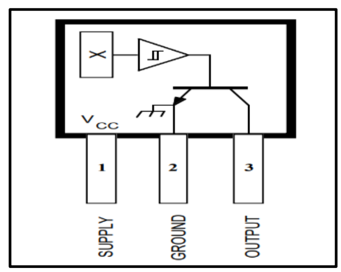

Figure 1.2: A3144 schematic

I2C LCD 1602

Displays the computed information: instantaneous speed and cumulative distance. The attached I2C module increases the simplicity of the hardware connection with the Arduino **(Fig. 1.3)**.

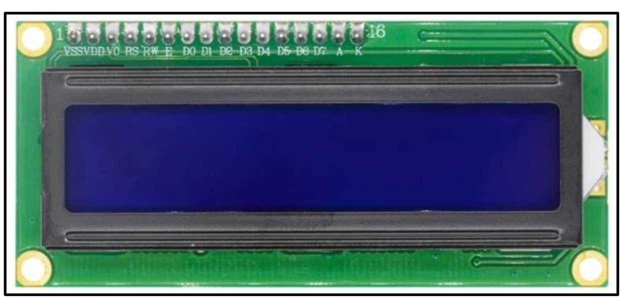

Figure 1.3: I2C LCD 1602 (Front-view)

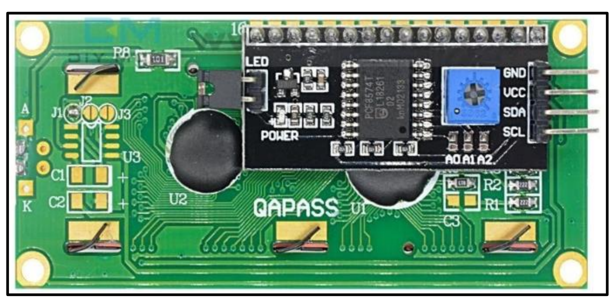

Figure 1.4: I2C LCD 1602 (Back-view)

UNDERLYING MECHANISM

The Hall effect sensor is fixed on the seat stay and the neodymium magnet is fastened onto a spoke **(Fig. 2.1)**. As the wheel rotates, the proximity between the magnet and the sensor decreases.

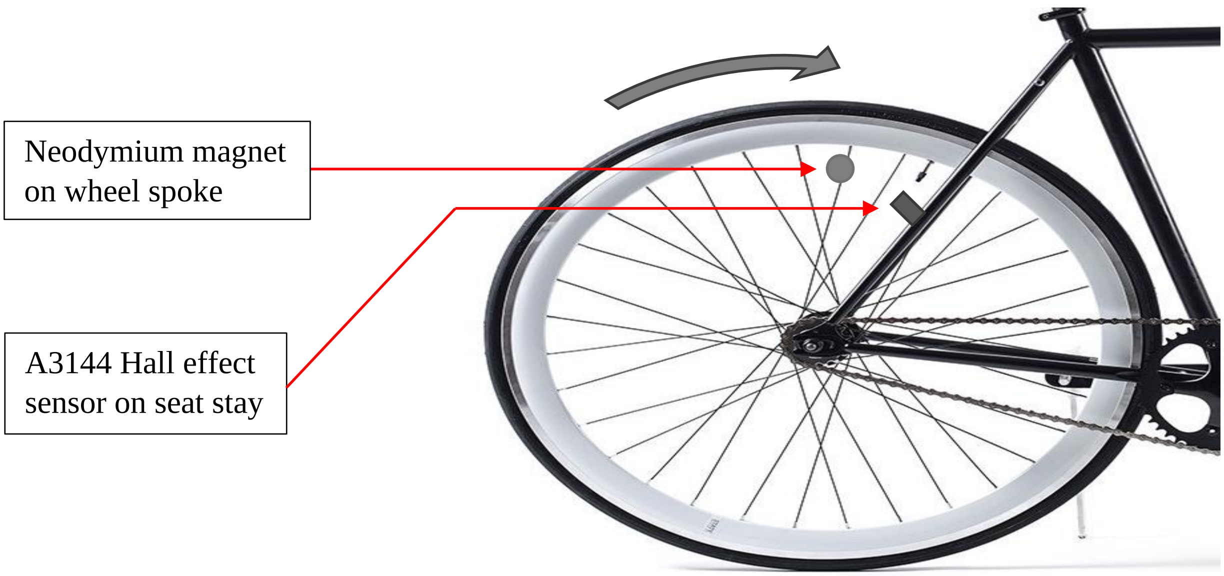

Figure 2.1: Bicycle with attached electronic essential components

Upon exceeding the threshold distance, the sensor actuates the circuit for a brief moment and generates an output voltage (Hall Voltage, V_H) or binary data as 1.

As the revolving neodymium magnet moves across the face of the Hall element, the sensing configuration is "sideways detection" **(Fig. 2.2)**.

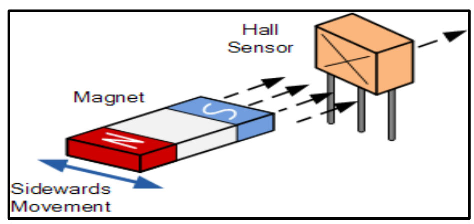

Figure 2.2: Hall element side movement

The Arduino calculates the total number of 1s inputted from the sensor per unit time. As the number of 1s is equal to the number of wheel rotations, a tachometer is designed. To reappropriate the tachometer into a speedometer and odometer, the wheel circumference was considered to convert rotational frequency into speed and distance.

The I2C LCD 1602 was fixed at the front of the bike to display the computed information: instantaneous speed and cumulative distance travelled.

CIRCUIT DIAGRAM SCHEMATIC

The electronic speedometer and odometer system developed in KiCAD describes the functions of the electrical components **(Fig. 3.1)**.

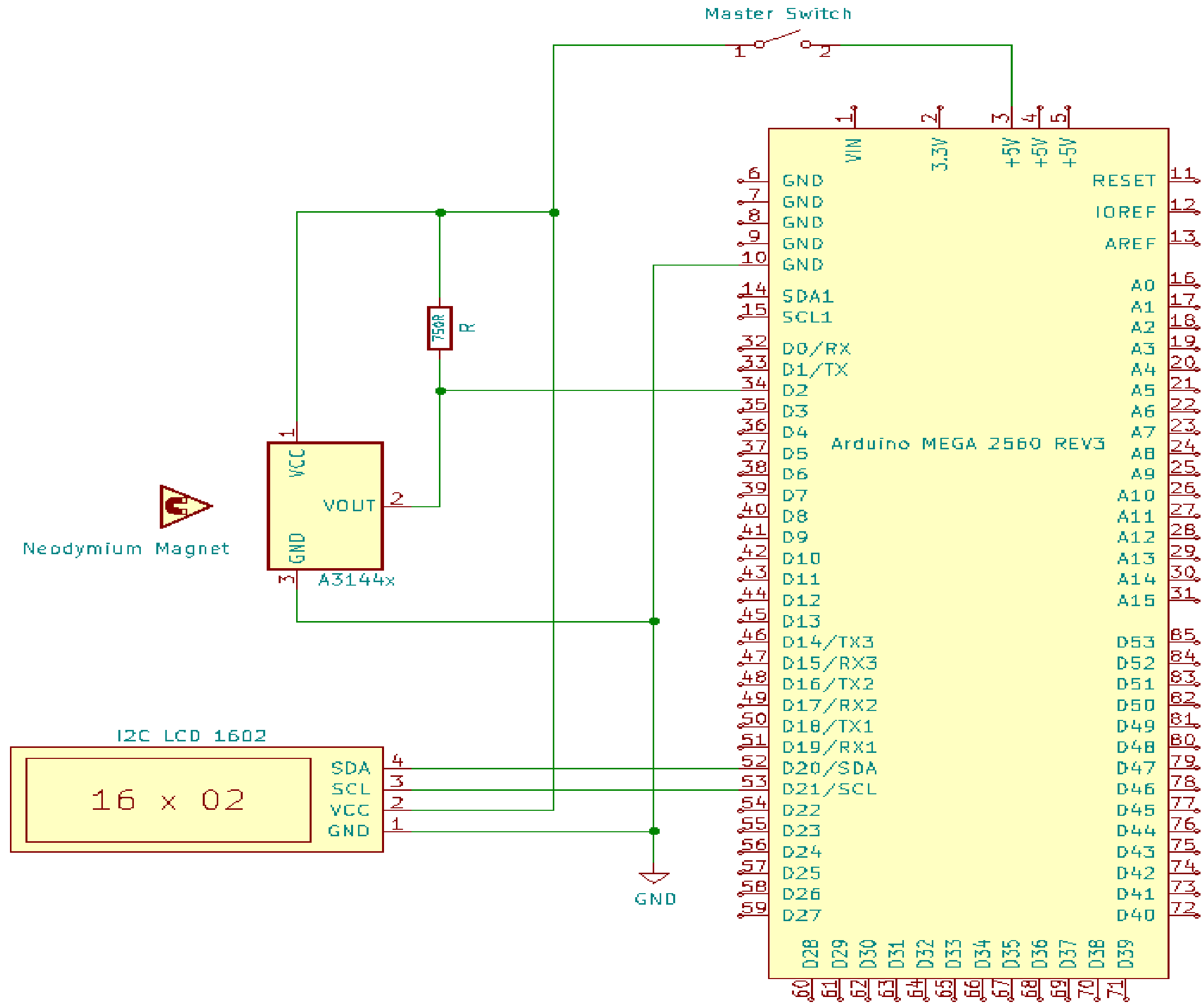

Figure 3.1: Speedometer and odometer electronic schematic (KiCAD)

Flow of Data & Information

Binary digital data from the A3144 is transmitted to pin D2 from V_{OUT}. Binary data in 0s and 1s correspond to the magnet's presence by virtue of whether the magnetic flux exceeds the pre-set threshold of the A3144.

The Arduino computes the data and transmits the processed data via. SDA and SCL pins to the I2C LCD 1602 for display. The established baud rate is 9600.

SOFTWARE DESCRIPTION OF SYSTEM

The full Arduino Sketch program (found on GitHub) is responsible for the following actions: receive, compute, and retransmit information.

Below showcases 2 of the program's essential functionalities:

Instantaneous Speed Calculation

To measure the time taken for 2 rotations of the wheel (magnet), the `millis()` function was utilized.

```
void magnet_detect()   //Function is called whenever a magnet is detected
{
  rotation++;               //Total number of rotations is incremented by 1
  dtime = currentTime;     //Time taken for the magnet's second detection
  if(rotation >= 2) {
    timetaken = currentTime-pevtime; //Time taken in ms for 1 full rotation
    rpm = 60000/timetaken;           //Formulae to calculate rpm
    pevtime = currentTime;
    rotation = 0;
  }
}
```

Given that the `timetaken` is a period and RPM is a frequency, the following conversion was conducted:

$$\text{RPM} = \frac{1}{timetaken} \cdot \left(1ms \cdot \frac{1s}{1000ms} \cdot \frac{1min}{60s}\right)^{-1}$$

$$\text{RPM} = \frac{60\ 000}{timetaken}$$

```
void loop()
{
  v = radius_of_wheel * rpm * 0.37699; //Formulae to calculate speed in km/h
}
```

Circumference and RPM units are meters and minutes, respectively:

$$speed = circumference_{wheel} \cdot RPM \cdot \left(1m \cdot \frac{1km}{1000m}\right) \cdot \left(1min \cdot \frac{1hr}{60min}\right)^{-1}$$

$$speed = 2\pi \cdot radius_{wheel} \cdot RPM \cdot \left(1m \cdot \frac{1km}{1000m}\right) \cdot \left(1min \cdot \frac{1hr}{60min}\right)^{-1}$$

$$speed = radius_{wheel} \cdot RPM \cdot 0.37699$$

Cumulative Distance Calculation

To design an odometer, the following formulae was constructed:

```
void magnet_detect() //Called whenever a magnet is detected
{
  rotationTotal += 1;    //Total number of rotations is incremented by 1
  distanceTotal = 2*rotationTotal*3.14*radius_of_wheel; //Total distance formula
```

Diameter unit of the bicycle wheel is in meters:

$$\text{distance}_{\text{total}} = rotation_{total} \cdot circumference_{wheel}$$

$$\text{distance}_{\text{total}} = rotation_{total} \cdot 2\pi \cdot radius_{wheel}$$

FINISHED PRODUCT

Picture of electronic components with breadboard: