REPORT OF INDIVIDUAL DESIGN PROJECT

INTAKE 39

PORTABLE BICYCLE SPEEDOMETER/ODOMETER WITH GPS TRACKER

BY

M. A. E. WIJESURIYA

D/ENG/22/0120/ET

BACHELOR OF SCIENCE IN ENGINEERING IN ELECTRONICS & TELECOMMUNICATION

GENERAL SIR JOHN KOTELAWALA DEFENCE UNIVERSITY

2024

ABSTRACT

This report outlines the design, development, and deployment of a GPS-based Odometer / Location Tracker device targeted at amateur cyclists, with the aim of providing an affordable, efficient solution that requires minimum setup and technical knowledge to operate. This system aims to provide a reliable, low-cost device that can be used to both display speed and distance travelled by the bicycle, as well as transmit location data to a server where it can be read in real-time. The core components utilized include a GPS transceiver module for receiving GPS signal data, an Arduino Uno microcontroller for processing and controlling the modules, two OLED displays to display information in real time, and a GSM/GPRS transmitter to transmit the data over the internet, along with a power supply for the whole system. Special attention was given to utilize low-cost and power-efficient devices, so that the device could be distributed and assembled with ease. Additionally, eco-friendly materials were utilized in making the casing, highlighting the importance of sustainability in engineering design. Aside from the hardware components, a website was set up to function as the frontend of the server storing the received location data, which could be viewed by users in the form of a user-friendly map. The device aims to encourage cycling as both a form of exercise and recreation, and an environmentally friendly means of transportation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

In today's modern and technologically advanced civilization, the average individual uses some form of mechanical transportation almost daily, sometimes travelling large distances for various purposes, be it business, leisure, health or numerous other reasons which are vital to keep society as a whole functioning correctly. With the development of these personal transportation systems, petrol-based (and diesel-based) automobiles, such as cars and motorbikes, have emerged as the most commonly used form of transport globally [1]. However, though these forms of transport are a testament to humanity's ability to create complex powered vehicles and transportation systems, they are not without their share of problems.

Vehicles, namely automobiles, have two main problems associated with them and their infrastructure. The first is the issue of traffic congestion, which can be seen in all major cities across the globe. Traffic congestion leads to wasted time, irritated commuters, and of course, blocked streets. The second, and possibly more impactful issue, is the issue of environmental pollution. Vehicles that utilize internal combustion engines, in addition to being extremely inefficient in terms of power consumption, are responsible for emitting many different pollutants and gases, including carbon monoxide, and nitrogen and sulfur oxides, all of which have drastic negative impacts on the environment. In recent years, these problems have led to many commuters finding alternative modes of transport. For example, during the economic crisis in Sri Lanka in the year 2022, many commuters started utilizing bicycles as a mode of transportation due to the shortage of petroleum-based fuels in the country. [2]

When cycling on main roads that are also utilized by motor vehicles, however, cyclists must exert caution and remain alert and aware of their surroundings and themselves. The more information that is available to cyclists, the less danger there is of meeting with an accident. One such variable that can be very useful to cyclists is their speed at any given moment. Knowledge of real-time speed has been shown to reduce the risk of vehicular accidents. [3]

Another variable that is useful to cyclists, particularly those who cycle for exercise and competitively, is distance travelled over a particular period of time. This data helps them to set goals for themselves and track their progress towards such goals. In addition to the simple numerical data, a visual representation of the geographical route travelled is both helpful and visually appealing to said cyclists. Taking these three factors into consideration (Speed, Distance, and Route-Tracking) the Cyclometer was designed.

CHAPTER 2: PRODUCT SURVEY

Cyclocomputers and distance trackers are by no means novel concepts, yet the majority of the products available on the market fall firmly into either of two categories which we will now define here, High-End (those products which are mainly targeted at professional cyclists and cost upwards of 200 USD) or Budget (products which are more suitable for the ordinary cyclist, costing within a price range of around 20 – 60 USD). In this section we examine the features and drawbacks of products falling into both these categories closely.

2.1 BUDGET PRODUCTS

The products in this category have prices ranging from around 20 USD to 60 USD

SIGMA BC 8.0 WL

This product uses a Hall Effect sensor that is attached to the rim of the bike wheel, which is used to count the number of revolutions the wheel makes in a given time. From that data, the speed the bike is travelling at is calculated. This information is then displayed on a large LED screen. [4] However, this system has a few drawbacks. Firstly, as the diameter of the wheel size is a factor in these calculations, this device has to be calibrated for each bike that it is attached to. This means transferring it from one bike to another is not a simple task. Additionally, the accuracy obtained from this system is not as accurate as data obtained from a device like a GPS receiver.

Figure 1: Sigma BC 8.0 WL Cyclocomputer [4]

SIGMA BC 8.0 WR

The wired version of the Sigma BC 8.0 WL, this product is similar in function to it, aside from the added drawback that communication between the device and the sensor is carried out through a wired connection, instead of a wireless connection. [5]

CATEYE QUICK WIRELESS CYCLING COMPUTER

This product also has an easy to read display and wireless communication, but also utilizes a wheel sensor that requires calibration and no other features. [6]

Figure 2: Cateye Quick Wireless Cycling Computer [6]

2.2 HIGH-END PRODUCTS

These products all cost upwards of 200 USD and are mainly marketed towards professional cyclists.

GARMIN EDGE 830

This product utilizes a GPS receiver for tracking and speed calculations. It also provides the cyclist with statistics, performance monitoring and many other features. However, many of these features are unnecessary for all but professional cyclists. Additionally, the cost of this device is upwards of 400 USD, which is higher than the average cost of a smartphone in Sri Lanka. [7]

Figure 3: Garmin Edge 830 [7]

SIGMA ROX 12.1 EVO

Similarly to the Garmin Edge 830, this product utilizes a GPS receiver as its main input sensor from which it calculates all other information. It also uses Wi-Fi connectivity, and many other smartphone app integrations as well. However, it suffers from many of the same drawbacks as the Garmin device, being priced at 360 USD. [8]

From the surveying of these products on the market, we can see that both types have their drawbacks, with Budget cyclocomputers tending to utilize Hall Effect sensors for their calculations, and High-End cyclocomputers being overfilled with features and carrying a price tag that is not suitable for the common cyclist. Therefore the proposed device will aim to fill this gap.

CHAPTER 3: OBJECTIVES AND AIMS

Objective – To design, develop, and implement a GPS-based cyclocomputer utilizing low-cost components, offering real-time speed and distance tracking to the user, as well as server-based route mapping.

Aim - To empower cyclists with a low-cost device that provides real-time performance data and route visualization for better training and exploration, encouraging eco-friendly, healthier transportation.

CHAPTER 4: METHODOLOGY

The project of developing a cyclocomputer was chosen due to its potential to conveniently track speed and distance during cycling. Although this is possible by using devices such as a smartphone, in practical situations this can pose a physical risk to the device as it is usually mounted on the front of the bicycle. Therefore a purpose-built bicycle speedometer provides a more robust solution, mitigating risks to personal electronic devices.

4.1 PRODUCT AND LITERATURE REVIEW

Before initializing the design of the device, existing products such as those mentioned above and literature on the topic was reviewed. It was discovered that the majority of existing cyclocomputer utilize one of two input devices in their designs:

- A Hall Effect sensor [4] [5] [6] or
- A GPS module [7] [8]

After weighing the benefits and drawbacks of both types of input modules and researching extensively into their potential accuracies and configurations, it was decided to utilize the GPS module design when building this cyclocomputer for two main reasons:

- Greater accuracy of GPS modules over Hall Effect sensors [9] [10]
- Availability of low cost GPS modules such as the Neo 6M module [11]

From this literature and product review, it was able to be satisfactorily concluded that while numerous bicycle speedometers exist, many rely on wheel-based sensors like Hall Effect sensors. These sensors are generally less expensive but can be prone to inaccuracies due to factors like tire slippage or varying terrain. GPS-based speedometers, on the other hand, tend to be more accurate but also carry a higher cost. This project aims to develop a GPS-based bicycle speedometer while prioritizing cost-effectiveness. The goal is to provide cyclists with reliable speed and distance data without the premium price associated with many existing GPS speedometers.

With these factors in mind, the sourcing and planning of the hardware for the product design was initialized.

4.2 HARDWARE

To begin planning the system out, a flowchart with the desired outcome and function of the device was prepared. The following flowchart was proposed initially:

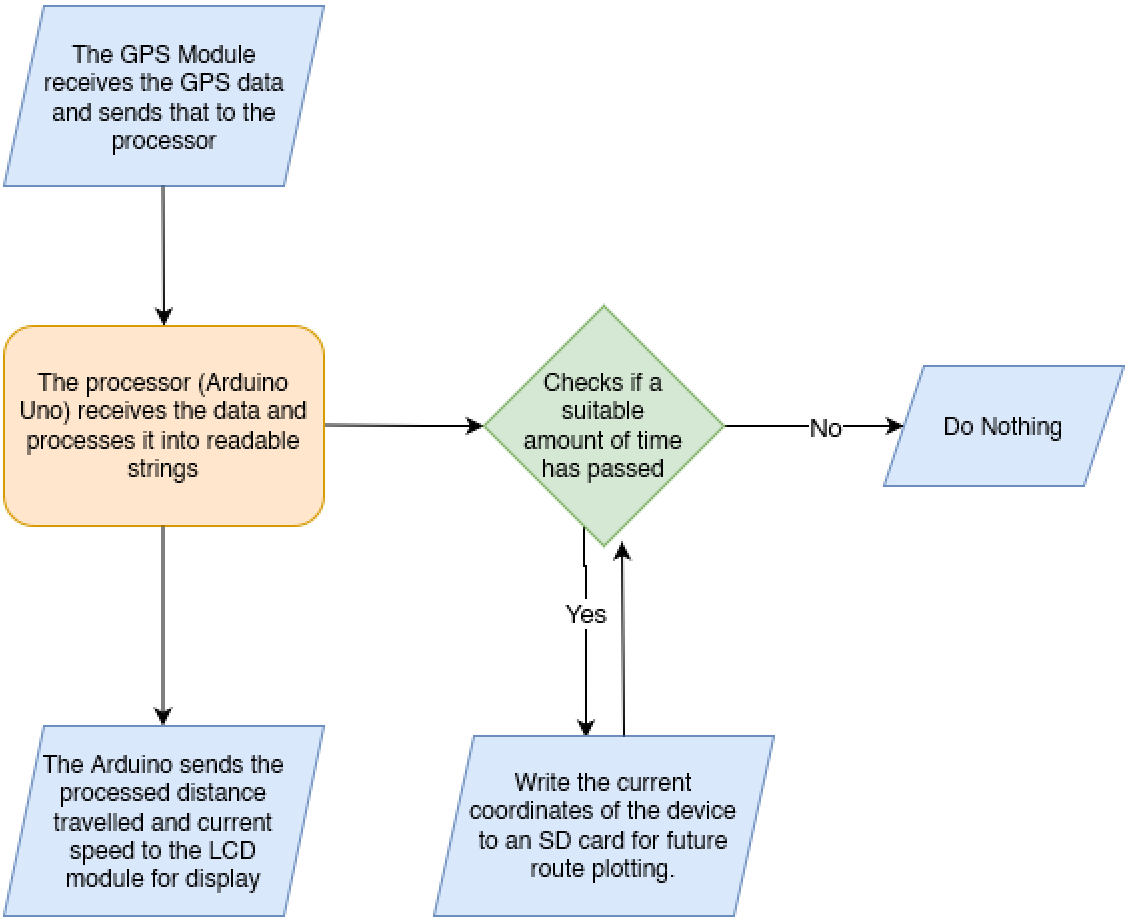

Figure 4: Flowchart describing the System Design

The hardware initially proposed for the device included:

- **Arduino Uno -** for processing data and controlling the other modules
- **Neo 6M GPS module w/ antenna-** for receiving the raw GPS data
- **SD Card Module** – to record the GPS coordinates for future route tracking
- **LCD Display module** – for displaying the speed and distance travelled to the user in real-time
- **9V battery cell** – for power supply to the system

The Arduino would receive the raw GPS data and save the latitude and longitude coordinates periodically to a file on the SD card in a CSV format. This CSV file could then be imported into a map software (e.g.: Google Maps) in order to display the route travelled by the user.

4.2.1 CHALLENGES WITH HARDWARE

However, after the initial proposal to the panel, it was suggested that the route tracking be done through some wireless means of communication for ease of access for the user, as well as real time route tracking. Therefore after deliberation and research, it was decided to replace the SD card module with a wireless GPRS module, in order to transmit the GPS data wirelessly over a data connection. For this purpose, the SIM800L GSM/GPRS module was chosen.

During the design and setup of the device, it became apparent that a 9V battery would not be sufficient to power all the modules required for this device, a fact which was amplified by the addition of the SIM800L module, which required 2A of current during peak operation. For this reason, an 18650 Battery Shield V8 that utilized two 18650 battery cells was chosen to replace the 9V Battery. A DC-DC Boost converter that stepped the 5V output to 7.5V for usage with the Arduino Uno's V_{in} pin was also used for this purpose.

Additionally, the LCD display was replaced by two Adafruit SSD1306 OLED displays, in order to gain visibility as well as save power and program memory.

Finally, after research, it was decided to use a 5V to 3.3V level shifter for communication with the SSD1306 displays as well as the NEO 6M GPS module, due to the unreliability of the supplier specifications. As the original modules were designed to run at 3.3V [12], it was considered safer to utilize a 3.3V signal instead of communicating directly with the Arduino Uno's 5V logic levels. [13] [14] [15]

Therefore the final component list for the device was as follows:

- **Arduino Uno Rev3**
- **Neo 6M GPS Module with Antenna**
- **2 x SSD1306 OLED 128x64 Display**
- **SIM800L GSM/GPRS Module**
- **18650 Battery Shield V8**

- **4-Channel 3.3V 5V Bi-Directional Logic Level Shifter Module**
- **DC to DC Adjustable Step-Up Power Supply Module**

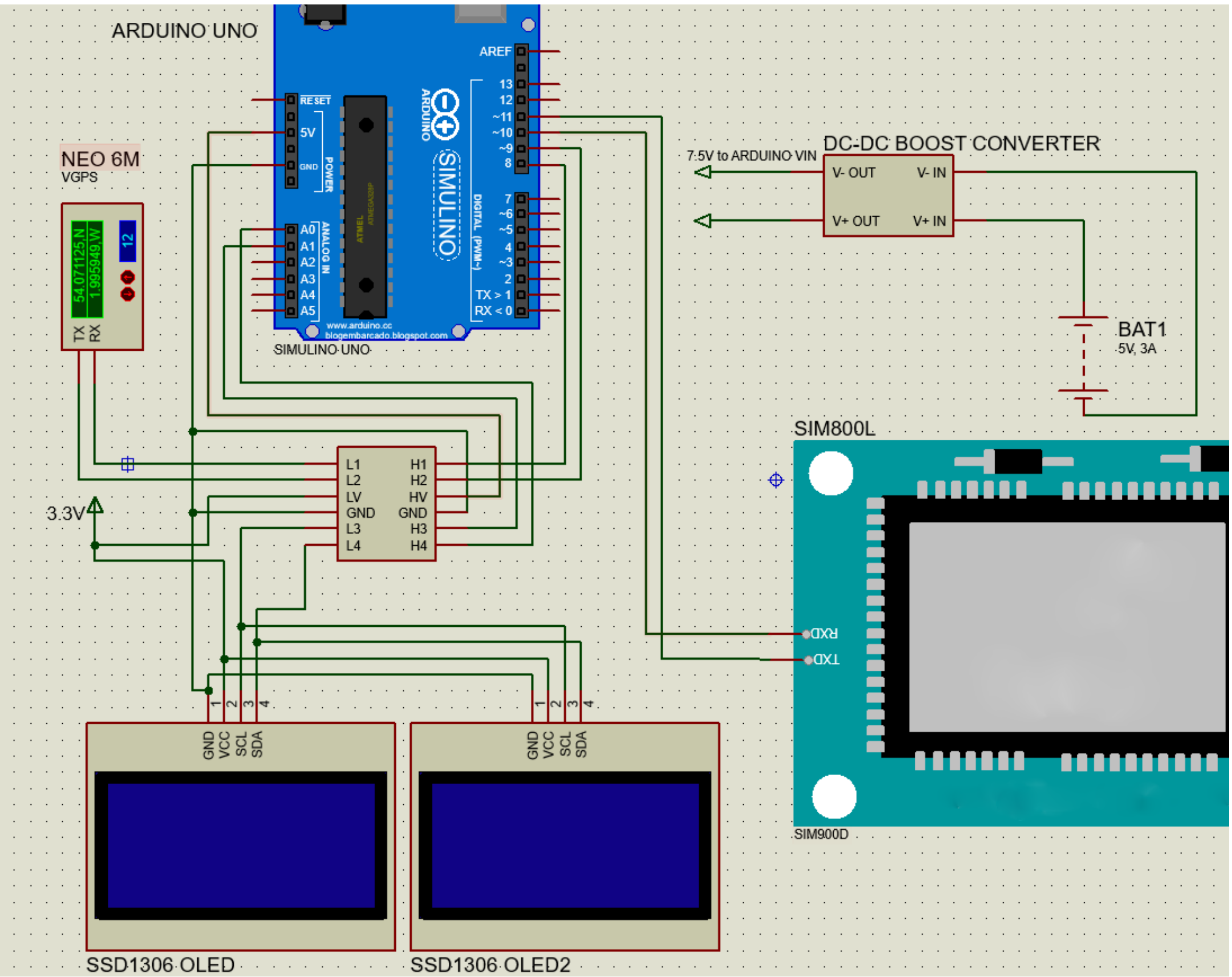

Figure 5: Proteus Illustration of Circuit Connection

4.2.2 HARDWARE

ARDUINO UNO

The Arduino Uno Rev3 was chosen as the microcontroller for this device due to its affordability, ease of programming, and the vast assortment of libraries available for it. The libraries utilized for the programming of this device include:

- TinyGPS++ - for reading and parsing NMEA strings from the GPS module
- SoftwareSerial – to allow serial communication on digital pins of the Arduino Board
- U8X8 – for controlling and displaying on the two OLED screens

The program was adapted and optimized for successful operation within the constraints of the 32KB of program memory and 2KB of SRAM [16] available on the Arduino Uno.

SSD1306 OLED DISPLAY

Two SSD1306 128x64 OLED Displays were used to present the ‘Speed’ and ‘Distance Travelled’ information to the cyclist in real time. These displays were chosen for their affordability, visibility/ high-contrast under strong lighting situations, and low power draw (0.04-0.06W at normal use [17, 18]). The displays use I2C serial communication and therefore require only two pins connected to the Arduino. Since the device uses 3.3V logic for TTL, a 5V to 3.3V level shifter was utilized.

As I2C devices require different addresses in order to control multiple devices from one controller, the physical jumper located on the back of one of the displays was soldered in order to change its address from 0x3C to 0x3D. The two displays were then soldered onto a PCB board with jumper wires for connection to the Arduino Uno. The U8X8 library was used to control the displays in favour of the Adafruit SSD1306 library and the U8g2 library. This was chosen due to its reduced program space requirements as it is a driver for text display only, eliminating space taken by unnecessary animation and graphical elements. [19]

NEO-6M GPS MODULE

The NEO-6M GPS module was chosen as it is a well-documented, cost-effective GPS module which is readily compatible with microcontrollers such as an Arduino Uno. This module uses an RS232 TTL interface for communication. It includes a built in backup battery and built in EEPROM. This module originally utilized an external U.FL ceramic antenna, however this was replaced by an enclosed, weatherproof, 1575.42MHz GPS Antenna for better reception. [20]

The operating voltage of the NEO-6M module is 3.3V, therefore the 5V to 3.3V level shifter was utilized for communication with the Arduino.

SIM800L GSM/GPRS MODULE

The SIM800L module is a Quad-band 850/900/1800/1900 MHz GSM/GPRS module that utilizes a TTL serial interface compatible with either 3.3V or 5V [21]. This module was chosen as a cost-effective, reliable means of communicating over wireless networks. A Hutch SIM card was utilized for this module, allowing it to communicate over the 2G mobile network. The data received from the GPS was transmitted over this device to the server encapsulated as a JSON string.

18650 BATTERY SHIELD V8

The 18650 Battery Shield was chosen for its many features and peripherals, such as dual Micro USB/ USB C charging inputs, USB A output + 5x 5V/2A output pins + 5x 3V/1A output pins, overcharging and overheating protection [22]. The 5V output pins were employed to reduce device size, offering a more compact power delivery solution than a USB A connector.

DC-DC ADJUSTABLE STEP-UP POWER SUPPLY

A DC-DC Boost converter was utilized in order to convert the 5V output from the 18650 battery module into a 7.5V output suitable to drive the Arduino Uno through the V_{in} pins [23]. A module capable of handling over 2A current was chosen, to ensure that the Arduino Uno would receive sufficient current.

4-CHANNEL 3.3V 5V BI-DIRECTIONAL LOGIC LEVEL SHIFTER MODULE

A bi-directional level shifter module was chosen to shift the Arduino Uno logic levels from 5V down to 3.3V for communication with both the two SSD1306 displays as well as with the NEO 6M GPS module. A 4-channel module was used to provide two channels for each device.

4.3 CASING DESIGN

Initially it was decided to use a prefabricated default housing for the device, but after deliberation, it was decided to use a custom designed casing to reduce size and create a lightweight, compact design suitable for mounting on the handlebars of a bicycle. Aerodynamic factors were also taken into consideration, but due to time and resource constraints were unable to be fully realized in the functional prototype. Various methods of casing manufacture were considered, including epoxy resin casting and CNC laser cutting, but finally 3D printing was decided on due to increased design detail, affordability, access to 3D printing facilities, and sustainability of source materials (PLA Plastic).

4.3.1 CAD DESIGN

SolidWorks was utilized to design the casing, taking into consideration the space requirements necessitated by the Arduino Uno and Battery module due to their dimensional significance among all the components. The base of the enclosure consisted of a 12 cm x 12 cm square. The lid featured a sloped design, ranging in height from 5 cm to 3 cm. Two rectangular 0.96 inch apertures were incorporated into the lid's upper surface to accommodate the two OLED screens. Additionally, two smaller apertures (7.7 mm diameter) were positioned on the rear surface to mount the GPS and GPRS antennas. Mounting points for component attachment were integrated into both the base and lid, and fillets were applied throughout the model for both aesthetic appeal and ergonomic considerations. Two apertures were also added on the side to enable USB-C and Micro USB charging, but due to the limitations of 3D printing had to be manually reshaped.

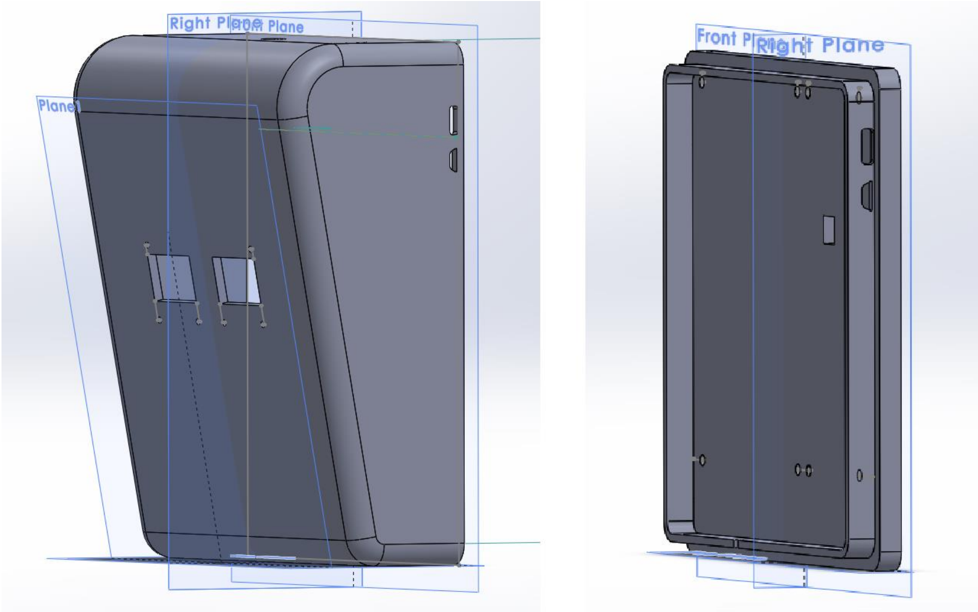

Figure 6: Isometric view of Lid (Left) and Base (Right) in Solidworks

4.3.2 PRINTING

The design of the models for the base and the lid were exported separately from Solidworks as two .stl files. These files were then imported into Ultimaker's Cura for slicing and printing. Printing was carried out using a Creality Ender 3 V2 SE 3D printer with 1.75mm green PLA filament. The lid was printed out at a layer height of 0.4mm while the base had a layer height of 0.3mm in order to prioritize printing speed over design intricacy. The recommended infill settings were used in order to maintain structural integrity within the casing and to ensure that it was able to house all the required components securely while being mounted on a bicycle.

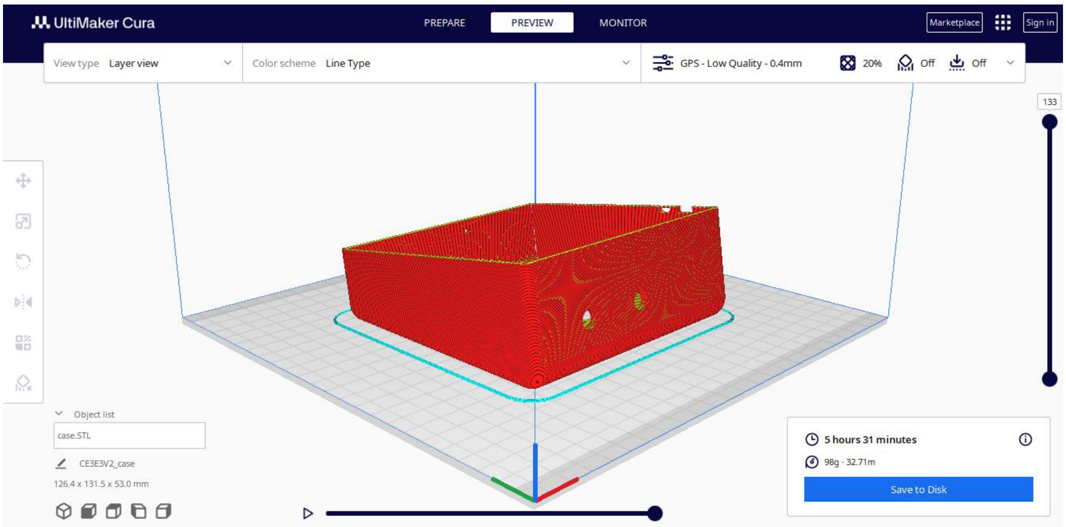

Figure 7: Enclosure Lid sliced in Ultimaker Cura

4.4 SOFTWARE

4.4.1 GPS CODING

The Arduino programming utilized the Software Serial to give commands to the NEO 6M and SIM800L modules as well as the U8X8 library to drive the two OLED displays.

```
U8X8_SSD1306_128X64_NONAME_SW_I2C u8x8(A1, A0, U8X8_PIN_NONE);
U8X8_SSD1306_128X64_NONAME_SW_I2C u8x82(A1, A0, U8X8_PIN_NONE);

#define rxSIM 6
#define txSIM 5
#define rxGPS 4
#define txGPS 3
//defines the TX and RX pins of the GPS and SIM module
SoftwareSerial sim800L(txSIM,rxSIM);
SoftwareSerial ss(txGPS, rxGPS);
```

Figure 8: Module Initialization in main program

The two OLED screens were initialized with one being programmed to display the Speed while the other displayed the Distance Travelled. Display fonts were chosen based on both readability and memory used.

```
u8x8.setI2CAddress(0x78);
u8x8.begin();
u8x82.setI2CAddress(0x7A);
u8x82.begin();
u8x8.setPowerSave(0);
u8x82.setPowerSave(0);

u8x8.setFont(u8x8_font_open_iconic_thing_2x2);
u8x8.drawGlyph(2,0,76);
u8x8.setFont(u8x8_font_saikyosansbold8_u);
u8x8.drawString(4,1,"DISTANCE:");
u8x8.drawString(7, 6, "KM");


u8x82.setFont(u8x8_font_open_iconic_thing_2x2);
u8x82.drawGlyph(2,0,81);
u8x82.setFont(u8x8_font_saikyosansbold8_u);
u8x82.drawString(5,1,"SPEED:");
u8x82.drawString(7, 6, "KM/H");
```

Figure 9: OLED Display initialization

Functions were coded to calculate the speed by taking the average of the speed received from the GPS in the $GPVTG sentence. [24] The average of the speed over ten cycles of the program was considered, in an effort to reduce naturally occurring GPS jitter. Additionally, the Distance Travelled variable was calculated by constantly updating the variable with the distance between the GPS coordinates taken at the time of calculation and the previous set of coordinates. This was combined by a conditional that discarded the values if greater than a threshold value (in this case 2 km) for the situation where a coordinate was mistakenly read as 0.00, thereby creating an invalid distance value.

```
if (cycles%1000 == 0 )
{
  double distanceKm =
  (double)TinyGPSPlus::distanceBetween(gps.location.lat(), gps.location.lng(), INIT_LAT,INIT_LON) / 1000.000;
  if (distanceKm > 2.00)
  {
    distanceKm = 0;
  }
  distanceDisp = distanceDisp + distanceKm;
  INIT_LAT = gps.location.lat();
  INIT_LON = gps.location.lng();
}
```

Figure 10: Function for Distance Travelled calculation

```
void calcSpeed(){
  for (int i = 0; i < 10; i++){
    speed = speedAvg[i] + speed;
  }
  speed = speed/10;
}
```

Figure 11: Function for Speed calculation

A function was created to update the displays with the updated variables every 500ms. In addition to this, a function was created to display the time received by the GPS in the top right corner of the screen.

```
if (millis() - lastUpdateTime >= 500) {
  calcSpeed();
  updateScreen();
  updateTime();
  lastUpdateTime = millis();
  speed = 0;
}
```

```
void updateScreen(){
  speed = gps.speed.kmph();
  u8x82.setFont(u8x8_font_profont29_2x3_n);
  u8x82.setCursor(4,3);
  u8x82.print(speed);
  u8x8.setFont(u8x8_font_profont29_2x3_n);
  u8x8.setCursor(4,3);
  u8x8.print(distanceDisp);
}

void updateTime(){
  u8x82.setFont(u8x8_font_saikyosansbold8_u);
  u8x82.setCursor(11,0);
  u8x82.print(gps.time.hour() + 5);
  u8x82.setCursor(13, 0);
  u8x82.print(":");
  u8x82.setCursor(14, 0);
  u8x82.print((gps.time.minute() + 30)%60);
}
```

Figure 12: Functions for updating the Displays and Time

4.4.2 GPRS/SIM CODING

For the SIM800L module, AT commands were used extensively to send the SMSs and send POST requests to the server over HTTPS. Initially a set of commands were sent to initialize the module with the network, as shown below

```
void waitForResponse(){
  delay(1000);
  while(sim800L.available()){
    Serial.println(sim800L.readString());
  }
  sim800L.read();
}

void simSetup(){
  sim800L.println("AT");
  waitForResponse();
  sim800L.println("ATE1");
  waitForResponse();
  sim800L.println("AT+CMGF=1");
  waitForResponse();
  sim800L.println("AT+CNMI=1,2,0,0,0");
  waitForResponse();
}
```

Figure 13: AT commands for initialization of SIM800L

Next, a set of AT commands were used to send an SMS as well as an HTTP POST request containing the acquired location data to a preconfigured server. The SIM800L documentation was utilized extensively in order to formulate the set of necessary commands to carry out this function. [25]

```
void send_sms(){

  sim800L.println("AT+CMGF=1"); // Configuring TEXT mode
  waitForResponse();
  sim800L.print("AT+CMGS=\"+94715990981\"\r");
  waitForResponse();

  sim800L.print("Lat:");
  sim800L.print(gps.location.lat(), 6);
  sim800L.print("\nLon:");
  sim800L.print(gps.location.lng(), 6);
  sim800L.write(0x1A);
  waitForResponse();
}
```

Figure 14: AT commands for sending SMS with coordinate data

```
void send_data_gprs() {
  // 1. GPRS Setup
  sim800L.println("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"");  // Set GPRS connection type
  waitForResponse();
  sim800L.println("AT+SAPBR=3,1,\"APN\",\"hutch3g\"");
  waitForResponse();
  sim800L.println("AT+SAPBR=1,1"); // Activate GPRS bearer
  waitForResponse();

  // 2. HTTP Request Setup
  sim800L.println("AT+HTTPINIT");
  waitForResponse();
  sim800L.println("AT+HTTPPARA=\"CID\",1");
  waitForResponse();
  sim800L.println("AT+HTTPPARA=\"URL\",\"https://cyclometer.000webhostapp.com/getdata.php\"");
  waitForResponse();

  sim800L.println("AT+HTTPPARA=\"CONTENT\",\"application/json\"");
  waitForResponse();

  //3. Build JSON payload
  String payload = "{\"lat\":";
  payload += gps.location.lat();
  payload += ",\"long\":";
  payload += gps.location.lng();
  payload += "}";
  sim800L.println("AT+HTTPDATA=" + String(payload.length()) + ",5000");
  waitForResponse();
  sim800L.println(payload);
  waitForResponse();

  // 4. Send Request and Close
  sim800L.println("AT+HTTPACTION=1");  // 0 for GET, 1 for POST
  waitForResponse();
  sim800L.println("AT+HTTPTERM");
  waitForResponse();
```

Figure 15: AT commands for sending HTTP POST requests

Initially Webhook.site was utilized for testing purposes, to ensure that the SIM800L was transmitting data to the server correctly. [26]

4.4.3 SERVER CONFIGURATION

Several methods for visualizing the cyclist's route were considered during the project's conceptualization. The initial strategy involved utilizing the Google Maps API, given its widespread adoption and the robustness of Google's infrastructure. Unfortunately, a researched revealed the absence of an API within Google's suite that could directly facilitate the desired route visualization [27]. For this reason this approach was discarded, and an alternate solution was planned.

For the alternate solution, a Wordpress site was created. Wordpress was chosen due to its ease of use, popularity, large library of plugins, and free hosting services provided by many companies 0 [28]. The Waymark Wordpress plugin was used to display the cycling route on the webpage [29] [30].

The server worked in the following way:

- The SIM800L module transmits the Latitude and Longitude coordinates provided by the NEO 6M GPS module as JSON container to a preconfigured PHP file stored on the server.
- The PHP file reads the incoming HTTP POST request and extracts the Latitude and Longitude values from the JSON string as variables.
- The PHP file creates a new GeoJSON file if no HTTP request has been received in the last 5 minutes, renaming the old file with a timestamp, or continues writing to the existing file if there has been an HTTP request within the last 5 minutes
- The PHP file then appends the new coordinates to the GeoJSON file as part of a LineString geometry object.
- This GeoJSON file is configured to be displayed as a map on the website.

The GeoJSON was structured using the documentation provided by the Internet Engineering Task Force (IETF) [31].

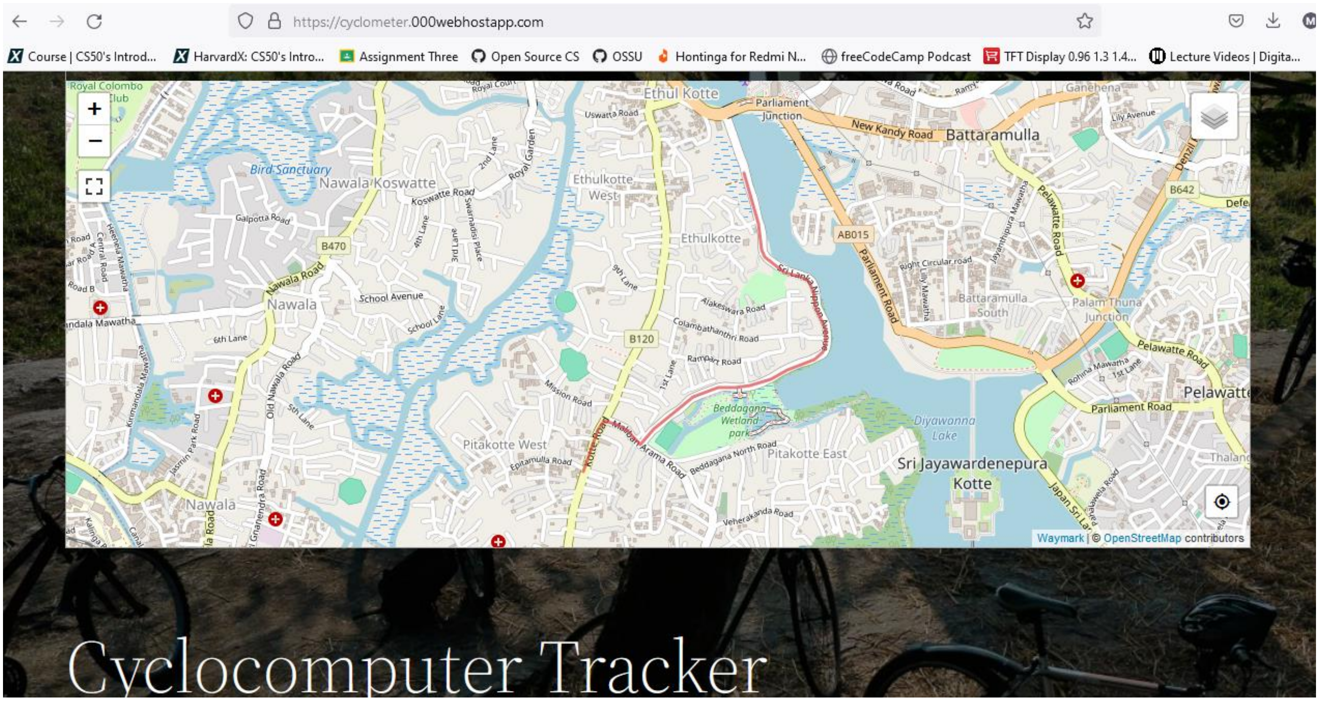

Figure 16: Front Page view of the website used for route tracking

```python
import requests
import json
import time

url = 'https://cyclometer.000webhostapp.com/getdata.php'

lon = [73.90263, 73.90297, 73.90322]
lat = [6.88834, 6.88925, 6.89005]
i = 0
n = len(lat)
while i < n:
    # Define the JSON data
    json_data = {
        "lon": lon[i],
        "lat": lat[i]
    }
    json_string = json.dumps(json_data)

    try:
        # Send the POST request with the JSON data
        response = requests.post(url, json=json_data, timeout=10)
        if response.status_code == 200:
            print("Request successful!")
            print("Response:")
            print(response.text)
        else:
            print("Request failed with status code:", response.status_code)

    except requests.exceptions.Timeout:
        print("Request timed out.")

    i += 1
```

Figure 17: Python script used to test server by sending pseudo location values

```php
<?php
// Set the directory where geoJSON files are stored
$directory = 'geojson_files/';

// Check if data is received via POST method
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get the JSON data sent from the Python script
    $json_data = file_get_contents("php://input");
    echo "Received JSON data: $json_data\n";

    // Decode the JSON data into an associative array
    $data = json_decode($json_data, true);

    if ($data !== null) {
        // Process the decoded JSON data
        // You can access individual elements using array keys
        $lon = $data['lon'];
        $lat = $data['lat'];
```

Figure 18: Extract from PHP file used to handle HTTP requests with location data

CHAPTER 5: FINAL OUTCOME AND DISCUSSION

Following complete assembly, the speedometer's functionality was tested within a motor vehicle in order to be able to fine-tune certain parameters of the code and to ensure proper functionality of both the speed and distance travelled calculations. After the completion of testing, the GPS-based speedometer was mounted to a bicycle using brackets and screws and its functionality was demonstrated to the panel during the final presentation.

Figure 19: Top (Left) and Side (Right) views of the completed device

The device primarily functioned as designed; however, the SIM800L module demonstrated intermittent connectivity to the server, with approximately a 50% failure rate. These failures appear to be due to issues within the HTTP connection establishment phase.

Additionally, due to the limitations of the 3D printing and the prototype design, the designed on/off switch did not function as expected to, requiring manual switching from inside the casing.

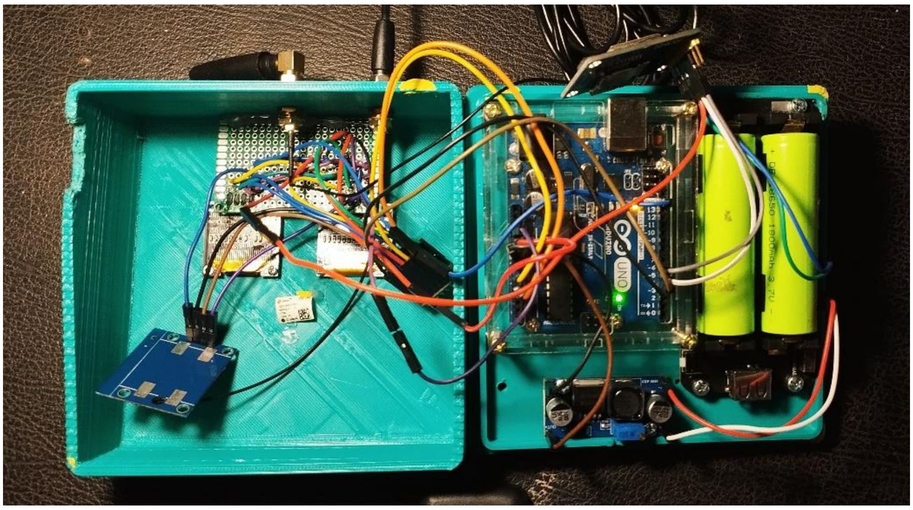

Figure 20: Internal wiring of the circuit component

5.1 PROBLEMS OVERCOME

Throughout the design, testing and deployment phases of the project, there were many issues and bugs that required troubleshooting and in some cases adaptation of the code to circumvent the issue. The following are a few examples of such scenarios.

- **Low Visibility of screen** – As mentioned earlier in this report, the initial component for the display of values to the user was a 16 x 2 LCD screen due to its ready availability and ease of programming. However, during testing in an outdoors environment, it was discovered that the LCD display provided very low visibility in bright sunlight.

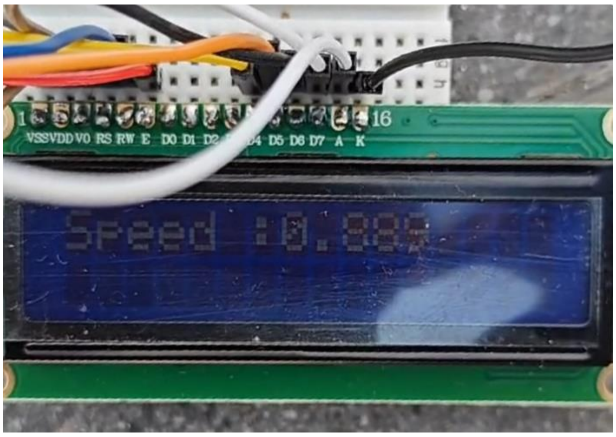

Figure 21: Low Visibility of LCD Screen

Therefore it was decided that this would pose a safety hazard to cyclists straining to read the display, causing them to be distracted and potentially leading to road accidents. After some deliberation, it was decided to split the display between two 0.96” OLED Displays, compromising on size (as no other size of OLED display could be reliably sourced for a suitable price) but gaining a significant improvement in visibility, especially in direct sunlight conditions. Additionally, it provided the dual advantage of requiring both less power consumption, as well as less physical pin space.

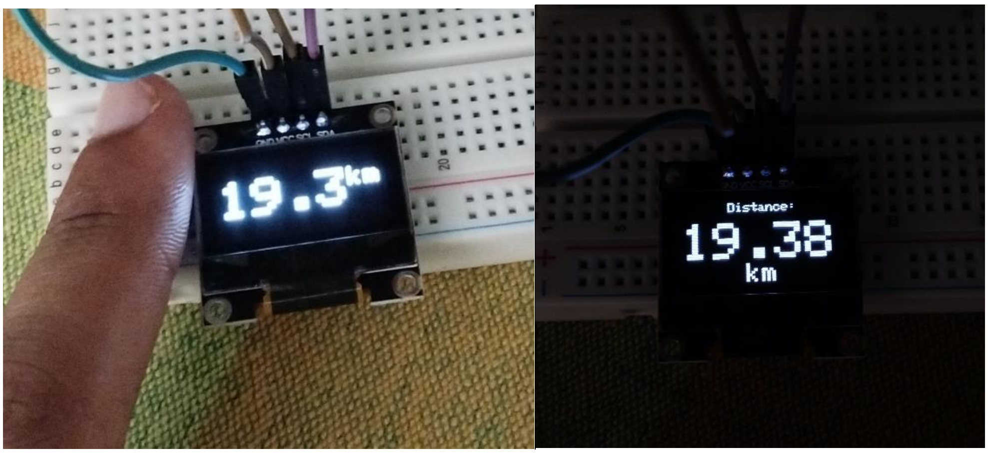

Figure 22: Demonstration of the High Contrast of the OLED Display

- **OLED Display Issues –** Since both OLED displays utilized the I2C protocol, unique addresses were necessary to prevent conflicts. To resolve the issue of default address overlap (0x3C), one display's I2C address pads were modified through delicate soldering, changing the I2C address to 0x3D, enabling the use of both devices on a single clock and data line. Additionally, while initially using the Adafruit SSD1306 library, its high RAM usage (1kb per display) proved unsuitable for the Arduino's limited SRAM (2kb) [23]. Extensive research led to the adoption of the U8X8 library, which provided the required functionality with much less RAM utilization, effectively addressing the memory constraints.

- **Route Tracking method –** Initially, a SD card was used to store CSV files with location data that could later be imported into a route tracking software for visualization. However, it was felt that this was a lot of unnecessary work for the user. Therefore a wireless network transmission system was proposed, with a link to Google Maps API for route tracking.

  After much research, however, it was noted that Google Maps' API suite lacked a suitable API that was able to both track, display and store route information. This was a significant setback, as the presence of such an API would streamline the server side process of route visualization. Eventually, after alternative solutions were considered, a WordPress site with adaptable plugins was developed, with a custom built PHP script to handle all the functions that was previously hoped to be handled by the API. Knowledge of PHP coding, GeoJSON file formats and WordPress plugins was required to be gained for this implementation.

- **AT Command Issues –**The SIM800L module, as a GPRS module, offers a wide range of complex and useful functions. However, navigating the extensive AT command library can be challenging, with many commands and functions seemingly overlapping, leading to certain AT commands not being supported by certain models of GPRS modules. This led to many trial and error experiments with various different combinations of AT commands before eventually arriving at an optimal set of functional commands for sending HTTP POST requests to a server.

5.2 STRENGTHS

Although the device is still not fully complete, the final product has many advantages over the other products available on the market at preset. The following are a few.

- **Costing -**

Table 1: Cost of the Project[a] *[11] [18] [20] [21] [32]*

Component	**Unit Cost (Rs.)**	**Total Cost (Rs.)**
Arduino Uno	1230	1230
Battery Holder V8	1500	1500
18650 Battery x2	650	1300
SSD1306 OLED Display x2	560	1120
NEO 6M GPS Module	1200	1200
SIM800L GPRS Module	2200	2200
1575MHz Antenna	750	750
DC-DC Boost Converter	350	350
4-Channel Level Shifter Module	200	200
Jumper Cables	150	150
PCB Board	150	150
3D Printer Filament (250g)	4200	1050
Total		**11200**

[a] - The majority of the parts were sourced from the store tronic.lk. The cost associated is the pricing provided by this store.

Component costs, as outlined in the table above, total Rs.11,200. Factoring in projected labor expenses, the estimated retail price would approximate Rs.14,000. This positions the device within the budget cyclocomputer market segment as defined previously (20-60 USD [6] [4] [5]), with pricing comparable to existing products (e.g., 46.92 USD at publication [33]). Furthermore, bulk procurement of components has the potential to reduce production costs significantly. The absence of comparable, domestically-produced alternatives means that this product is able to substantially reduce import costs and associated tariffs, typically 20% of the consumer cost [33], providing a competitive cost advantage within the local cyclocomputer market.

- **Availability -** Additionally, all components listed here are readily available in most electronics stores, making the design of this speedometer easily accessible to many cyclists. This enables ordinary cyclists to have access to GPS-based speedometer tracking and speed calculation.

- **Portability -** Due to the compact size of the device, transferring from one bicycle to another is made possible by the utilization of standardized handlebar mounts using standard screws. The GPS antenna utilizes a magnetic based adhesive system to affix itself to the handlebars. No further configuration is required, making the device user-friendly with its portability and ease of setup. This gives the device a significant advantage over devices utilizing Hall Effect sensors, which require careful calibration before initial use. [6] [4] [5]

- **Compatibility -** The use of standard mounts further ensures this speedometer device's compatibility with a wide range of bicycle types, opening up the possibility of adaptation for other vehicles such as trolleys and pushcarts. The utilization of PLA plastic for casing fabrication provides weather resistance for the device, allowing it to function in inclement weather situations as well.

- **Anti-theft Functionality -** In addition to providing functionality for the user, the GPS route tracking system also effectively combats theft by enabling covert location monitoring from the cloud. This real-time tracking provides valuable data in the event of theft, without alerting the perpetrator, enabling the stolen bicycle to be located and the thief deterred with ease.

5.3 LIMITATIONS

Over the course of the development of this project, much has been achieved and accomplished. However, due to both lack of resources and design inexperience, the final outcome still has many limitations and flaws. The following are a few of these.

- **Lack of Power Switch –** Due to the aforementioned limitations of the 3D printed casing, the location of the power switch on the battery pack, and the lack of a separate addressable power switch pin, there is no easily accessible on/off switch on the device, and as it is must be switched on from inside the casing. This is a design flaw which reduces the overall user-friendly aspect of the device. By connecting a physical switch through an aperture in the casing, this flaw can be eliminated.

- **Lack of protection for Displays –** The OLED displays are visible through the apertures in the lid of the casing. However, these apertures have no covering due to the lack of a suitable transparent material. This reduces the protection given by the casing and renders the device susceptible to inclement weather conditions. By adding custom made plastic transparent coverings for the apertures, this limitation can be addressed and minimized.

- **SIM800L Issues –** The SIM800L successfully transmits a HTTP POST request approximately 50% of the time. This is unsatisfactory for normal use, as an acceptable success rate should be ideally about 85-90%. The causes of this issue are difficult to determine and are likely due to many different causes, both on the server end and the device end.

- **Data Privacy/Security issues –** In its current iteration, the device transmits location data via standard HTTP POST requests to a publicly accessible web server. While this configuration suffices for demonstration purposes, a commercially viable product demands a secure and dependable solution for storing user data on private servers.

5.4 KNOWLEDGE ACQUIRED

Throughout each stage of designing and deploying this project, extensive knowledge was acquired about various fields of study. This knowledge has already proved useful for application in various other disciplines as well. Some of the topics discussed include:

- **Cyclocomputers –** The field of cyclocomputers in general, gaining an in depth understanding of the inner workings of cyclocomputers, the different types of hardware utilized to carry out computation, the features provided, and the software that is used in conjunction with them. This knowledge formed the basis of the planning for this device, and allowed for the selection of appropriate features and technologies to achieve the desired functionality. Additionally, it facilitated the discovery of a market gap to target when designing the product.

- **GPS NMEA Sentence Structure –** In order to decode the GPS NMEA sentences received as raw data from the NEO 6M GPS module, a basic understanding of the NMEA sentence structure was required. From these sentences, variables such as latitude, longitude, speed, and time were extracted and utilized for display.

- **OLED and LCD Display mechanisms –** For display purposes, both LCD and then OLED display modules were utilized. In order to understand how to drive and use these modules for text and graphical display, knowledge of the underlying electronics and programming needed to run the displays was needed. Display multiplexing and I2C addressing was also learnt as a by-product, as well as Arduino driver functionality and memory optimization. Substantial knowledge of the OLED display mechanism was especially gained, as the program was severely limited in terms of memory and maximum optimization was necessary. This led to research into font display types, screen buffering, and other memory optimization techniques. Both the Adafruit SSD1360 documentation and the U8X8 library documentation were invaluable in gaining this information. [12] [19]

- **Soldering Techniques –** Various components necessitated the use of soldering either header pins or wires for different connections. Therefore, a practical knowledge of effective soldering techniques and other knowledge such as recognizing the importance of strong electrical connections and their impact on circuit reliability were gained during the creation of this device

- **AT Command Syntax –** For the operation of the SIM800L, AT commands were required to be passed to the module to enable it to function correctly. This required an extensive knowledge of the AT command syntax, including commands to initialize the connections, set connection types, and containerize the HTTP POST data for transmission. This acquired knowledge has practical applications in various places within the field of Telecommunications. [25]

- **CAD and 3D printing –** In order to create the casing for the model, a custom designed case was required to be designed from the ground up and subsequently prepared for printing using a 3D printer. Basic knowledge of Computer Aided Design using

Solidworks, as well as slicing using software such as Ultimaker Cura was gained. Additionally, working within the constraints of the 3D printing facilities led to acquiring knowledge of optimization of print speeds and the various factors influencing print quality and the structural stability of 3D printed items. This knowledge also has potential application in a vast number of fields in the future, as the technology of 3D printing develops and evolves as time progresses. [34] [35]

- **Python and PHP Language –** For setting up the server that was used for route tracking, a PHP script (Fig. 18) was written to handle the incoming POST requests, and to process them into data readable by the maps plugin for visualization. This required knowledge of the PHP language. Additionally, a Python script (Fig. 17) was written to send test requests to the server to ensure the proper functionality of the visualization. This knowledge of programming is invaluable as it is required in practically every field of engineering in the present day.

- **GeoJSON file format and WordPress –** The Wordpress plugin required a GeoJSON file format in order to read and display the data on the site. Therefore it was required to process the data received by the PHP script into some form of GeoJSON geometry to enable the visualization of the route travelled. Knowledge of the GeoJSON file format, as well as the geometry structures contained within (specifically LineStrings, which are used to display routes) was necessary for this, and was gained from the official documentation for the GeoJSON format specified by the Internet Engineering Task Force [31]. The website used for display was created with WordPress for ease of use and availability of plugins. Therefore, a basic understanding of WordPress design and plugin functionality was needed.

CHAPTER 6: CONCLUSIONS AND FUTURE WORKS

Through the design, deployment and testing of this Portable Bicycle Speedometer / Odometer with GPS tracking, extensive knowledge in the field of electronics, telecommunications, programming, and general product design has been gained. Importantly, it addresses a market gap by providing a low-cost, GPS-based alternative to traditional, Hall-Effect based cyclocomputers. This empowers budget-conscious cyclists to access accurate tracking technology provided by GPS. Additionally, with features like route mapping and sharing capabilities, the device could become attractive to casual cyclists and commuters seeking user-friendly navigation tools.

By making cycling appealing in this manner, this device plays a major role in encouraging a healthier lifestyle, as well as providing incentives in the form of statistical data that can be used as a benchmark for improvement. In this way, this project also highlights the potential for electronics and GPS technology to play a role in promoting sustainable transportation and healthy habits.

Future improvements for this device would include a redesign of the casing, focusing on a more refined, detailed structure with support for a power switch, as well as more weather resistant designing, including a transparent shield for the OLED displays. On the server end, a secure method for individual users to gain access to their own route data could be designed, with attractive, easy to read UI displaying additional features such as Time Elapsed, Distance Travelled, etc. These improvements would further strengthen this device's position in the market by appealing to a wider user base of cyclists and filling in the market gap.

Overall, this device has the potential for providing a low-cost, GPS based solution for Bicycle Speed and Distance tracking for budget-conscious cyclists. This has the potential to benefit society in general by encouraging a healthier, alternate mode of transportation.

REFERENCES

[1] European Commission, Directorate-General for Mobility and Transport, "EU transport in figures – Statistical pocketbook," Publications Office of the European Union, Luxembourg, 2022.

[2] C. Beeler, "Amid fuel crisis in Sri Lanka, bicycling is no longer a 'poor person's mode of transport'," The World, 4 August 2022. [Online]. Available: https://theworld.org/stories/2022-08-04/amid-fuel-crisis-sri-lanka-bicycling-no-longer-poor-persons-mode-transport. [Accessed 14 March 2024].

[3] D. F. B. Flynn, A. Breck, O. Gillham, R. G. Atkins and D. L. Fisher, "Dynamic Speed Feedback Signs Are Effective in Reducing Driver Speeds: A Meta-Analysis," *Transportation Research Record*, vol. 2674, no. 12, pp. 481-492, 2020.

[4] Sigma Sport, "BC 8.0 WL - Sigma Sport," [Online]. Available: https://sigmasport.com/product/bc-8-0-wl/. [Accessed 21 March 2024].

[5] Sigma Sport, "BC 8.0 WR - Sigma Sport," [Online]. Available: https://sigmasport.com/product/bc-8-0-wr/. [Accessed 21 March 2024].

[6] Cateye, "Quick | Products | Cateye," [Online]. Available: https://www.cateye.com/intl/products/computers/CC-RS100W/. [Accessed 21 March 2024].

[7] Garmin, "Garmin Edge 830 | Cycling Computer," [Online]. Available: https://www.garmin.com/en-US/p/621232. [Accessed 21 March 2024].

[8] Sigma Sport, "ROX 12.1 EVO - Sigma Sport," [Online]. Available: https://sigmasport.com/product/rox-12-1-evo/. [Accessed 21 March 2024].

[9] R. Aisuwarya, M. A. Riyan and R. E. Putri, "Design of Bicycle's Speed Measurement System Using Hall Effect Sensor," *Journal of Physics: Conference Series*, vol. 1339, no. 1, pp. 12-18, 2019.

[10] O. D. S. H. R. N. B. e. a. Siddiqui, "Evaluating the Accuracy and Reliability of Bicycle GPS Devices," *SAE International Journal of Advances and Current Practices in Mobility*, vol. 3, no. 6, pp. 3093-3114, 2021.

[11] tronic.lk, " Ublox NEO-6M GPS Module (Small SMA Connector) (MD0561)," tronic.lk, [Online]. Available: https://tronic.lk/product/ublox-neo-6m-gps-module-small-sma-connector. [Accessed 24 March 2024].

[12] Adafruit, "SSD1306 Datasheet," April 2008. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf. [Accessed 8 April 2024].

[13] jess50, "SSD1306 OLED power supply 3V or 5V?," 7 November 2016. [Online]. Available: https://forum.arduino.cc/t/ssd1306-oled-power-supply-3v-or-5v/417350/2. [Accessed 8 April 2024].

[14] salmanaf, "Can my SSD1306 OLED use 5v I2C?," 21 March 2023. [Online]. Available: https://forum.arduino.cc/t/can-my-ssd1306-oled-use-5v-i2c/1104539. [Accessed 8 April 2024].

[15] seawarrior, "SSD1306 VCC, SCL & SDA voltages," 2 December 2020. [Online]. Available: https://forum.arduino.cc/t/ssd1306-vcc-scl-sda-voltages/685310. [Accessed 8 April 2024].

[16] B. Earl, "Memories of an Arduino," Adafruit Industries, 08 March 2024. [Online]. Available: https://cdn-learn.adafruit.com/downloads/pdf/memories-of-an-arduino.pdf. [Accessed 24 March 2024].

[17] Smart Prototyping, "OLED Display (0.96 in, 128x64, SSD1306, I2C)," [Online]. Available: https://www.smart-prototyping.com/OLED-0.96inch-12864-display-module-blue.html. [Accessed 24 March 2024].

[18] tronic.lk, " 0.96 inch 128X64 OLED Display Module White I2C," [Online]. Available: https://tronic.lk/product/0-96-inch-128x64-oled-display-module-white-i2c-iic. [Accessed 24 March 2024].

[19] olikraus, "u8x8reference," 19 May 2023. [Online]. Available: https://github.com/olikraus/u8g2/wiki/u8x8reference. [Accessed 8 April 2024].

[20] tronic.lk, "GPS Antenna 1575.42MHz," [Online]. Available: https://tronic.lk/product/gps-antenna-1575-42mhz-3v-5v. [Accessed 8 April 2024].

[21] tronic.lk, "SIM800L Quad-Band GSM GPRS module with Antenna V2.2 5V for Arduino," [Online]. Available: https://tronic.lk/product/sim800l-quad-band-gsm-gprs-module-with-antenna-v2-2-5v-. [Accessed 8 April 2024].

[22] diymore, "diymore 1865O Battery Holder Double Li-ion V8 Battery Shield Micro USB 5V/3A 3V/1A," [Online]. Available:

https://www.amazon.com/Diymore-Lithium-Battery-Charging-Arduino/dp/B07SZKNST4. [Accessed 8 April 2024].

[23] Arduino, "Arduino Uno Rev3 Documentation," [Online]. Available: https://docs.arduino.cc/hardware/uno-rev3. [Accessed 8 April 2024].

[24] M2MSupport.net, "Ground Speed from NMEA $GPVTG sentence," [Online]. Available: https://m2msupport.net/m2msupport/get-the-gps-ground-speed-from-the-nmea-gpvtg-sentence/. [Accessed 12 April 2024].

[25] SIMCom Wireless Solutions Limited, "SIM800 Series AT Command Manual," [Online]. Available: https://microchip.ua/simcom/2G/SIM800%20Series_AT%20Command%20Manual_V1.12.pdf. [Accessed 10 April 2024].

[26] webhook.site, "Webhook - Test, process, and transform emails and HTTPS requests," [Online]. Available: https://webhook.site. [Accessed 12 April 2024].

[27] Google, "Google Cloud Console," [Online]. Available: https://console.cloud.google.com/google/maps-apis/onboard?entry=devsite. [Accessed 12 April 2024].

[28] Hostinger, "Why use WordPress for your sitee," 26 March 2024. [Online]. Available: https://www.hostinger.com/tutorials/why-use-wordpress. [Accessed 12 April 2024].

[29] Waymark, "Waymark - Interactive Maps for WordPress," [Online]. Available: https://www.waymark.dev/. [Accessed 12 April 2024].

[30] M. Wijesuriya, "MW24 Cyclometer Tracker," [Online]. Available: https://cyclometer.000webhostapp.com/. [Accessed 12 April 2024].

[31] Internet Engineering Task Force, "The GeoJSON Format," [Online]. Available: https://datatracker.ietf.org/doc/html/rfc7946. [Accessed 12 April 2024].

[32] tronic.lk, " Arduino UNO Normal Development Board with USB Cable," [Online]. Available: https://tronic.lk/product/arduino-uno-normal-development-board-with-usb-cable. [Accessed 13 April 2024].

[33] Sri Lanka Customs, "Import Tarrifs - Sri Lanka," March 2024. [Online]. Available: https://www.customs.gov.lk/wp-content/uploads/2024/03/Chapter-91-1.pdf. [Accessed 13 April 2024].

[34] xe.com, "14,000 LKR to USD - Convert Sri Lankan Rupees to US Dollars," [Online]. Available: https://www.xe.com/currencyconverter/convert/?Amount=14000&From=LKR&To=USD. [Accessed 13 April 2024].