CHALMERS
UNIVERSITY OF TECHNOLOGY

Logging Data From E-Scooters To Improve Traffic Safety

Master's Thesis in Automotive Engineering

RAHUL RAJENDRA PAI

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se

MASTER’S THESIS IN AUTOMOTIVE ENGINEERING

Logging data from e-scooters to improve traffic safety

Rahul Rajendra Pai

Department of Mechanics and Maritime Sciences
Division of Vehicle Safety
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2022

Logging data from e-scooters to improve traffic safety

Rahul Rajendra Pai

Master’s Thesis 2022:14
Department of Mechanics and Maritime Sciences
Division of Vehicle Safety
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
Small fleet of Voi e-scooters with the data logger developed as a part of this thesis.

Chalmers Reproservice / Department of Mechanics and Maritime Sciences
Göteborg, Sweden 2022-08-17

Logging data from e-scooters to improve traffic safety
Master’s thesis in Automotive Engineering
Rahul Rajendra Pai
Department of Mechanics and Maritime Sciences
Division of Vehicle Safety
Chalmers University of Technology

Abstract

The transport sector has seen a major overhaul in recent years with the emergence of several new forms of transport. The increasing environmental concerns, fuel prices and traffic congestion have had people recon substitute means of transportation such as electric cars, e-bikes, e-mopeds etc. E-scooters due to the rapid growth of the scooter-sharing system have emerged as an attractive solution. These motorized scooters also help mitigate the first/last mile issue associated with the use of public transport. However, the introduction of any new mode of transport leads to new conflicts which thereby can result in new types of crashes. While the regulations in several parts of the world treat the e-scooters similar to bicycles, numerous studies have proven several dissimilarities in the overall dynamics between them. To understand the causes of conflicts, data about the usage of e-scooters is essential. Naturalistic data is data collected using instrumented vehicles in road traffic, by users performing their day-to-day activities. These datasets are less likely to suffer from bias as compared to data collected in a regulated environment such as a test track. Naturalistic data are a widely recognized source to analyze and model the behavior of road users, to improve traffic safety. Naturalistic data collected on e-scooters will provide a unique understanding of the details of e-scooterist riding behavior, interactions with the surrounding road users, and reactions in different situations. This thesis aims at developing a ride data logger which will enable naturalistic data collection when extended to a large fleet of e-scooters. In addition, preliminary data collection and analysis are conducted to test the effectiveness of the logger and identify the potential issues.

Data corresponding to 15 variables is logged at a frequency of 10 Hz from a plethora of sensors on the e-scooter, providing detailed information about the kinematics of the ride. In addition, video data is captured which provides a visual information of the riding environment and the rider. When the logger is connected to the internet the data stored in the local memory storage of the data logger is automatically transferred to the cloud storage. This not only automates the process but also minimizes human dependencies and interventions. To facilitate the analysis of the datasets collected as a part of the process, a Python-based graphical user interface, which allows visualization of each datapoint, has been developed. This help in analyzing the kinematics and video data simultaneously.

A pilot data collection involving participants has been carried out based on which a preliminary data analysis has been conducted to indicate the potential of the new data logger to serve for a large-scale naturalistic riding study. Exposing the prototype to different road surfaces, lighting conditions, riding styles, and ride durations over the span of 350 km has facilitated the identification of points of improvement for a furture large-fleet data collection.

Key words: Naturalistic Riding Study, Naturalistic Data, E-scooters, Data Logging, E-scooter safety, Vehicle and Traffic Safety, Naturalistic Data Collection

Contents

Preface

This thesis involved development of a data logger for e-scooters along with some preliminary data analysis. The development of the logger is a combination concepts from mechanical, electronics and computer engineering. Although it may appear complex, I have tried to explain it in a detailed manner but keeping in mind not to dilute the topic.

The Introduction chapter provides a comprehensive overview of the e-scooter phenomenon, naturalistic data collection and the previous work in this area. The chapter Methodology explains the complete process involved in hardware and software development. The subsequent section describes the preliminary data collection process to test the effectiveness of the setup developed. The data analysis chapter details on the data filtering followed by a few examples on how the data can be utilized in further studies. The final chapter discusses the findings along with the possible improvements in the future.

The solution is developed with scalability in mind. This allows it to replicate the setup onto a fleet of e-scooters which will thereby enable a large-scale naturalistic data collection.

Acknowledgement

First and foremost, I am extremely grateful to my supervisor Giulio for his immense support. Thank you for guiding me in every step, helping me in every possible way, believing in me throughout and more importantly making me feel like a family.

My deep gratitude to examiner Marco for his continuous support and inspiration during the thesis. Thank you for always steering the project in the right direction and providing valuable feedback throughout.

This thesis would have never been possible without André, my supervisor at Voi. Thank you for being my thesis partner more than my supervisor and taking a very active role in development of ‘Voiberry’ as you named it. Really appreciate the time you have dedicated to answer all my questions. All the discussions we had, helped me learn several new skills and technologies.

Special thanks to Daniel and Fredrik, the two musketeers who have been supportive throughout the thesis. Thank you for making my journey at Chalmers a memorable one. Tack så mycket!

I am grateful to Hardware BU at Voi Technology AB as a whole for providing a creative work environment. My thanks go especially to Mikael and Vigor, for taking part in discussions and actively contributing to the development of the prototype.

I thank Christian-Nils Boda for attending all the meetings and providing his valuable feedback at every stage of the thesis.

Hardware manufacturing would not have been possible without the support from Chalmers Solar Team especially the team members Mille, Christian, and Adil.

Special thanks to Ron Schindler and Fredrik Bruzelius for taking time and participating in the data collection.

I thank my parents and little brother who have always supported me in all my ventures with their unconditional love and blessings

Rahul Rajendra Pai, Göteborg, 2022-08-17

List of Figures

List of Tables

1 Introduction

1.1 Background

The ever-increasing fuel prices, serious environmental and health concerns over the use of combustion engine vehicles have had people recon substitute means of transportation. While electric cars help combat these issues, the exorbitant time spent in traffic congestion and soaring electricity prices make them a less viable option for shorter trips. E-scooters or as they call it in Swedish ‘elsparkcykel’, and other micro-mobility vehicles have been attractive alternatives. A study by Smith and Schwieterman [1] has shown that for trips between 0.5 to 2 miles, the e-scooters are the best substitutes for private vehicles. The ‘scooter sharing’ [2] companies can be attributed as the primary cause of the trend. With the introduction to the public on the streets of Santa Monica and San Francisco in 2017 [3], the e-scooter sharing companies now operate across various cities and provide users to ride and pay by the minute. The first-last mile dilemma has been a major hurdle for public transport, as compared to the private modes of transport [4]. These e-scooter sharing services aim to provide a solution to the first/last mile problem [5] with a report indicating that a major share of e-scooter users walk less than 50 meters to find an e-scooter [6]. Apart from the first-last mile dilemma, the electrically powered scooters with a battery range now as high as 55 km [7] have become an appealing mode of transport for even the slightly longer trips. Capable of running at speeds ranging from 20 to 25 km/h (varies based on regulations) these micro-mobility services are an attractive substitute for cars which in city limits run at an average of 15 to 20 km/h [8]. This is evident as a report by National Association of City Transportation Officials (NATCO) indicates that the trips on e-scooter across the United States increased from 38.5 million in 2018 to 86 million trips in 2019 [9], [10]. The popularity of e-scooters is not only shown by the increase in trips but also by the boost in scooter-sharing companies: over 40 different scooter-sharing companies operate across the globe with Europe alone having seen a rise of 19 operators in the span of 2 years [11]. Likewise, Sweden has seen an increase in e-scooter rental services with a total of 1.6 million users in 2021 alone [12]. In Sweden, the scooter-sharing companies started operations in 2018 [6] and are now available in 10 cities [13] with as many as 23000 e-scooters in the Stockholm region. As of 2021, there have been 9 operators (Voi, Bird, Bolt, Lime, Moow, Link, Hubb, Tier, Dott) across Sweden [14], [15]. Voi, one of the leading e-scooter rental companies has marked a peak demand of 300,000 trips in one day across 76 cities indicating the popularity of the e-scooter [16].

The e-scooters are currently being ridden in the bicycle lanes in Sweden. In places where the bicycle lanes are absent, the e-scooters are allowed to be ridden on the pedestrian walkways [17]. Although the bicyclist’s and the e-scooter rider’s behavior are non-identical [18], [19], the current traffic regulations treat them alike [17]. The increased number of users has led to new types of traffic conflicts with not only bicyclists and pedestrians [19] but also with other road users including heavy vehicles. With the rise in the number of riders, the topic of safety of e-scooter riders alongside the other road users is now even more important. With the e-scooter phenomenon on the rise, an increase in the number of e-scooter rider injuries and hospitalizations has been observed [20] with about 38 fatalities documented worldwide until 2019 [21]. A study conducted in New Zealand by Brownson et al. [22] indicates that 91.7% of the crashes were single crashes caused due to sliding, hitting the curbs etc. However, a study conducted by Stigson et al. [23] indicated about 83% of the reported e-scooter related injuries in Sweden were involved in a single crash. Various crash analyses have indicated that 6.4-8.7% of the crashes involved a pedestrian [24], [25]. The review conducted by Stigson et al. [24] reveals that nearly half of the reported crashes occur at night where environmental factors such as limited visibility may likely be a contributing factor. Studies also have reported about 54% of crashes to occur over the weekend, with multiple reports indicating intoxication resulting in crashes [25]–[27]. With respect to injuries, several e-scooter crash studies have indicated the recurrence of head injuries to varying degrees [20], [22], [23], [26], [28], [29].

From all the aforementioned research it is evident that there have been numerous safety concerns associated with the e-scooter use and several studies have been conducted on investigating the crashes of e-scooters. However, little to no research has been done to understand the crash causation mechanisms or the rider behavior that results in the crash, due to the lack of this type of information in crash databases. In order to avoid crashes or mitigate their consequences, it is critical to understand the events that lead to a crash and especially the e-scooterist's interaction with surrounding road users. Naturalistic data (ND) are a widely recognized source to analyze and model the behavior of road users, to improve traffic safety [30]. ND are data collected unobtrusively from road users driving or riding in the traffic, by equipping vehicles with cameras and sensors. Given the equipment installed on the vehicles is continuously recording the data at every point in time it provides us with the detailed behavior of the recorded road user before, during and after a conflict and not just limited to consequences as observed with crash databases. While the downside of ND collection is the limited crash data availability, there have been several studies that have indicated near-crashes to be a surrogate measure for crashes [30], [31]. Near crashes, as defined by Guo et al. [30] are safety-critical scenarios that require an instant evasive maneuver that otherwise would result in a crash. The research projects such as the 100-Car Naturalistic Driving Study, UDRIVE, euroFOT, and Strategic Highway Research Program-2 (SHRP-2) have mainly collected naturalistic data from motor vehicles [32]–[35]. Some studies have been carried out on bicycles [36]–[39] and in studies conducted on e-scooters, data are collected externally using video cameras mounted on cars and infrastructures [19], [40]. Collecting data from the perspective of the e-scooterist gives a different understanding as compared to the ones collected on cars or trucks. This is due to the limited interactions between the e-scooters and other motor vehicles, the data collected on the e-scooters can complement the data collected by other road users thereby increasing the addressable traffic situations.

1.2 Aim of the thesis

This thesis aims to contribute to a better understanding of the behavior of e-scooter riders and their interaction with other road users. The methodology used for achieving the aim is the collection of a pilot dataset from a Naturalistic Riding Study (NRS). The thesis will start with the development of a ride data logger and the data collection, focusing later on the data analysis. The dataset although may be relatively limited, will allow for preliminary analysis of the collected ND and provide a fundamental tool to understand the necessary improvements for conducting a larger NRS in the future.

1.3 Previous studies on data collection with e-scooters

Tian and Sherony [41] are currently working on the collection of real-world riding data of e-scooters by equipping cars and e-scooters. Data collected from cars done using a multicamera setup alongside a 3D LIDAR have been used to detect the e-scooters using computer vision technology [42], [43]. While some reports indicate the use of cameras and LIDAR on the e-scooters for data logging, there are no concrete details on how it is being done given this is ongoing research. Although to date this is the only ND collection on e-scooters, there have been some studies involving data loggers on e-scooters. One such study has been conducted by Violin [44] wherein he developed a methodology to understand the user behavior on various micro-mobility vehicles. As a part of the research, he equipped an e-scooter with a Raspberry Pi based logger running the Robotic Operating System (ROS) and recorded parameters such as braking, steering, acceleration etc. Given the e-scooter used here did not come equipped with the sensors the researcher equipped the e-scooter with a separate Inertial Measurement Unit (IMU), steering angle measurement and wheel speed measurement. While the logger has generated good results, it is restricted to be used in an experimentally confined area and not on commercial e-scooters.

A similar study has been conducted to specifically understand the kinematics and the dynamics of the e-scooter by Garman et al. [45]. They had equipped the e-scooter with a commercially available Plex VMU logger. In addition to the sensors similar to the ones used by Violin, Garman et al. have implemented a front-facing camera alongside the camera facing the foot of the rider. Using a similar data logger for naturalistic data collection can nevertheless prove to be expensive.

Brunner et al. [46] carried out a study to understand the dynamics and stability of e-scooter riders in turns. Similar to the logger developed by Violin, Brunner et al. included a screen along with a single board computer. As a part of the study, the researchers used an IMU to measure steering angle, velocity, and the leaning angle to get an idea of the stability of the e-scooterist giving hand signs while turning towards a specific direction. Identical to the two previously mentioned studies, the experiment by Brunner et al. has been carried out in a controlled environment which results in a deviation from the real-world data.

This thesis is a continuation of Automotive Engineering Project (AEP) work carried out by Schmidt et al. [47]. Schmidt et al. developed a prototype data logger for an e-scooter. The data logger was a Raspberry Pi 3 based unit coupled with a J-Link debugger and two camera modules. The prototype developed as a part of the project and the basic data flow is shown in Figure 1 and Figure 2 respectively. The J-link debugger extracts the data from the IoT of the e-scooter and sends it to the data logger. Kinematic information of the e-scooter is obtained from the sensors. The accelerometer data in addition to the wheel speed, GPS data and turn indicator signal are stored on a Universal Serial Bus (USB) stick. In addition, the front and rider cameras provide a visual representation of the riding environment and the rider’s face respectively.

Although it has been demonstrated that the prototype collected a reliable dataset there have been major critical issues that would prevent it from being adopted for a large-scale data collection. The prototype used J-Link debug probe to extract data from the sensors onboard the e-scooter which not only makes the setup bulky but also expensive. The setup had been powered with a power bank which meant that the participant had to charge it separately failing which data logging would not be possible. The unlock and locking operation had been carried out by means of a push-button which makes it more prone to theft. Optimization of the logger both in terms of hardware (to prevent vandalism and integration with the existing hardware) and data logging (data collection and management) is essential and is a key area of focus for this thesis.

Figure 1 Prototype developed by Schmidt et al. [47].

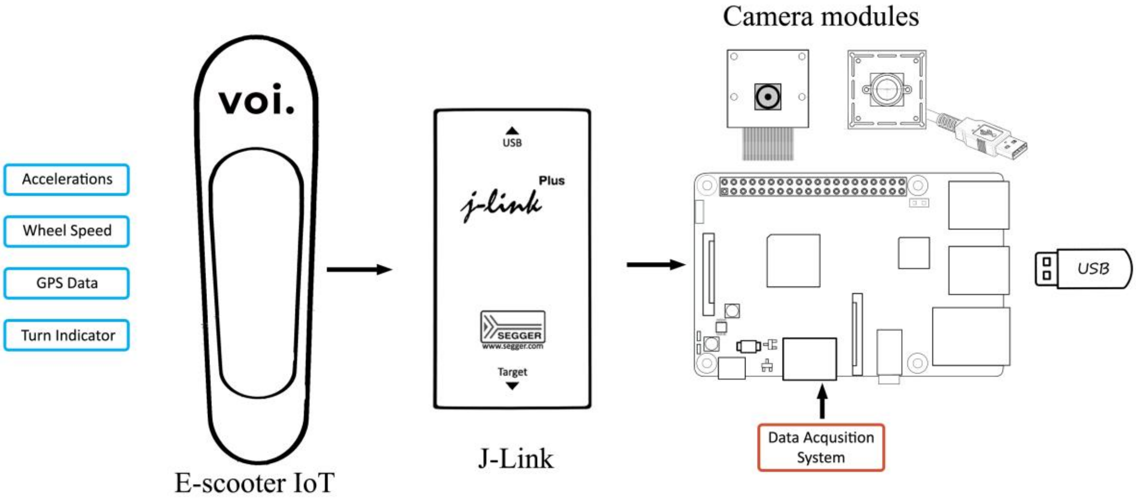


Figure 2 Data flow in the prototype developed by Schmidt et al. [47].

2 Methodology

2.1 Hardware

As mentioned in section 1.2, this thesis is a continuation of the AEP work by Schmidt et al. [47]. Therefore, the setup developed as a part of AEP (named as 'Prototype-0') served as baseline hardware. Given the upgrades required for the prototype, the overall data logger in this thesis has been developed in two stages with one prototype at each stage:

1. Optimization of the structure and dimensions of the logging setup.
2. Implementation of communication protocols into the logger and Voi mobile application-based logging functionality.

The first stage/prototype (named as 'Prototype-1') involved optimization of the structure of the data logger to enable a more compact and robust setup which is well integrated with the existing e-scooter. As shown in Figure 4 the data logger in the Prototype-1 is dimensionally optimized resulting in a compact setup as compared to the Prototype-0 shown in Figure 3. The Prototype-1 has two modules namely the logger and the J-Link on the front and rear of the stem respectively.

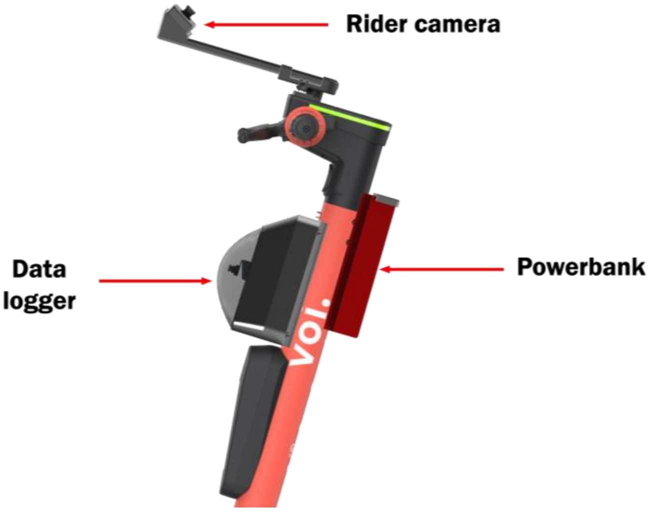


Figure 3 Prototype-0 developed by as a part of AEP by Schmidt et al. [47]

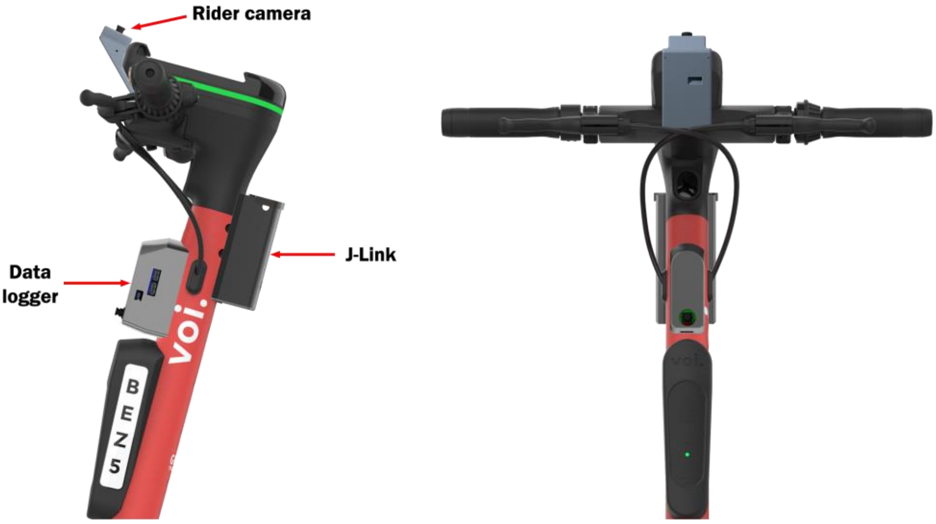


Figure 4 Prototype-1 e-scooter with dimensionally optimised hardware.

The major objective of the second stage/prototype (named as 'Prototype-2') has been to eliminate the dependency on the J-Link debugger. This involved setting up a UART based communication protocol resulting in the module on the rear of the stem being eliminated. The final setup on the Prototype-2 e-scooter is shown in Figure 5. To enable usage of the e-scooter like any other rental e-scooter, the unlocking of the e-scooter has to be carried out using Voi mobile application. Setting up of communication protocol has enabled achieving this target which has been an issue on both the Prototype-0 and Prototype-1. The following subsections provide detailed insight into each of the prototypes.

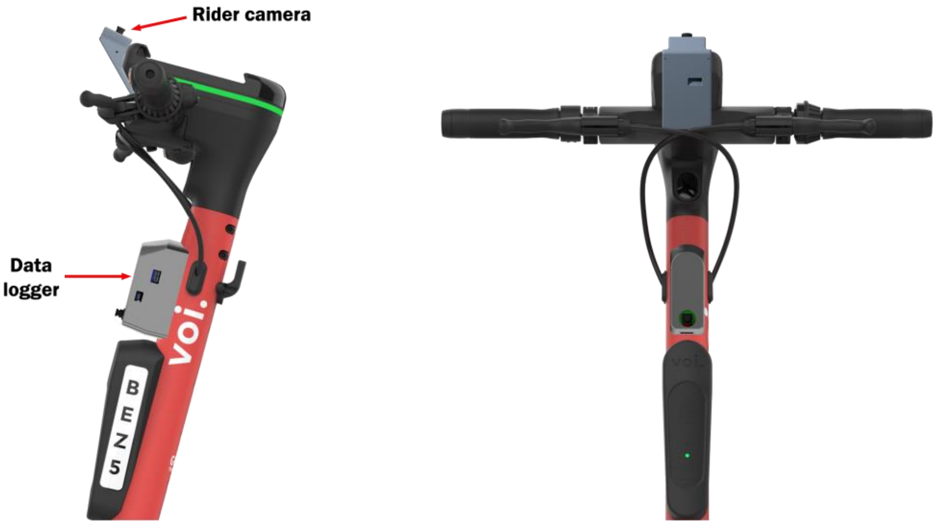


Figure 5 Prototype-2 e-scooter with communication protocol and mobile application-based logging.

2.1.1 Prototype-1

As mentioned in the previous section, the significant requirements of dimensional optimization and setting up of a communication protocol put forth on the data logger would result in little to no time for participant-based data collection. Hence the decision to develop the logger in two stages with each of the prototypes being as close to the other with no major hardware additions. This has enabled data collection with one of the prototypes while the subsequent one was under development. The Prototype-1 hardware comprises the following 10 parts:

1. Data logging computer
2. USB hub
3. J-Link
4. Hardware casing
5. Voltage converter
6. Trigger button
7. Lock/Unlock switch
8. Road-facing camera
9. Rider-facing camera
10. Camera mounts

2.1.1.1 Data logging computer and USB hub

When selecting the device to be used as a data logger, a single board computer with camera compatibility has been the main criterion. While the Prototype-0 used the Raspberry Pi 3B, the Raspberry Pi Zero 2W has a smaller form factor but retains the same ARM Cortex A53 chip as the Pi 3 [48], [49]. The side-by-side comparison of both the single board computers is shown in Figure 6. A USB hub shown in Figure 6B is attached over the Pi Zero 2W to connect the J-link and the USB camera. The J-Link module and the USB based rider-camera module are as shown in Figure 4 and are explained in detail in the following subsections of the report. Given the large-scale implementation of the data logger, the use of the Pi Zero 2 board helps reduce the cost of each setup by more than 50% [50], [51].

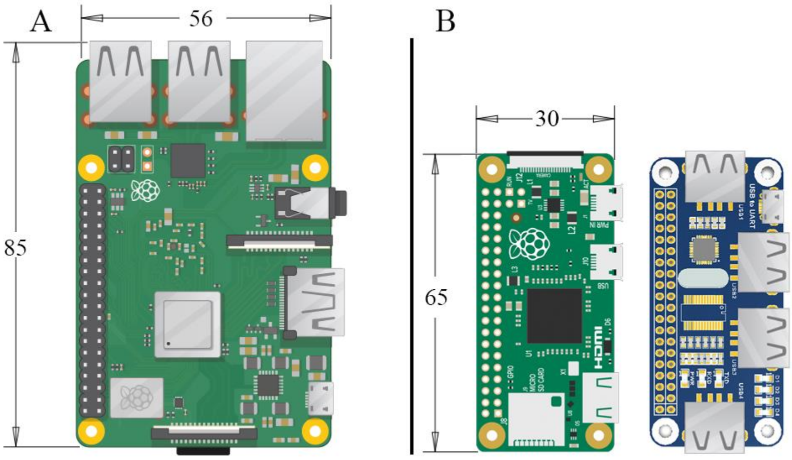


Figure 6 A) Raspberry Pi 3B and B) Raspberry Pi Zero 2W with the USB hub.

2.1.1.2 Hardware casing and camera mounts

Creating a robust and compact design that would look natural on the e-scooter while being simple to manufacture has been the major objective of the housing design. The design of hardware has been developed using PTC Creo [52]. For Prototype-1, the setup has been split into two modules that sit on either side of the stem of the e-scooter as shown in Figure 4. The module on the front includes the data logger along with the road-facing camera module while the module on the opposite side of the stem encases the J-Link debugger. The idea behind splitting the module into two is to develop a compact design and in the subsequent prototype eliminate the module at the back of the stem without any significant changes to the data logger module.

On the data logger module, the stacked USB hub and Pi Zero 2W assembly is placed on the baseplate with a spacer in between. The camera is mounted on the lower end of the Pi Zero by means of a camera mount. The voltage converter is placed inside a casing of its own on the lower end of the baseplate. With all components in place, the outer casing is placed on top and screwed from the bottom of the baseplate. The arrangement of each of the components in the data logger module is shown in Figure 7 and Figure 8. This assembled module is mounted on the e-scooter using the existing screw holes as shown in Figure 9. As shown in Figure 10, there is a significant decrease in the overall dimensions of the data logging module as compared to the Prototype-0.

The module on the back replaces the power bank module from Prototype-0 and houses the J-Link debugger. The lid for the module also houses the switch used to lock/unlock the e-scooter. The arrangement of each of the components in the data logger module is shown in Figure 11. This assembled module is mounted on the e-scooter using the existing screw holes as shown in Figure 9. As shown in Figure 12, the dimensions of this J-Link module are slightly smaller as compared to the power bank module used in the Prototype-0.

The rider camera is mounted around the phone holder on the e-scooter. Figure 13 shows the design of the camera mount on the e-scooter. The rider camera on the Prototype-0 has proven to be fragile resulting in bad videos when travelling on uneven surfaces due to vibrations. This has been corrected on Prototype-1. The Voiager-5 [53] has a sturdy phone holder at the center that helps add stability to the structure of the camera mount.

The case has been manufactured in-house by means of additive manufacturing at the eXPerimentverkstaden laboratory at Chalmers. Polyethylene terephthalate glycol (PETG) plastic has been used for this purpose. Although the material properties of Acrylonitrile Butadiene Styrene (ABS) are ideal for this application, it has not been used owing to the difficulty in manufacturing using the material.

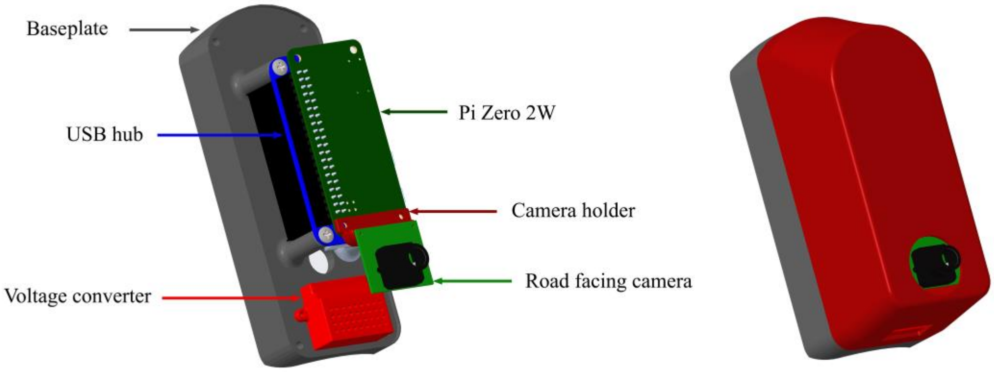


Figure 7 Isometric views of the data logger module.

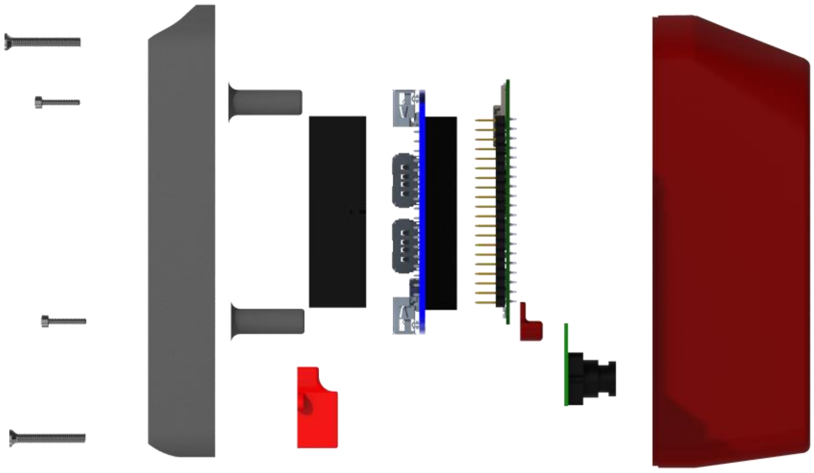


Figure 8 Exploded view of the data logger module.

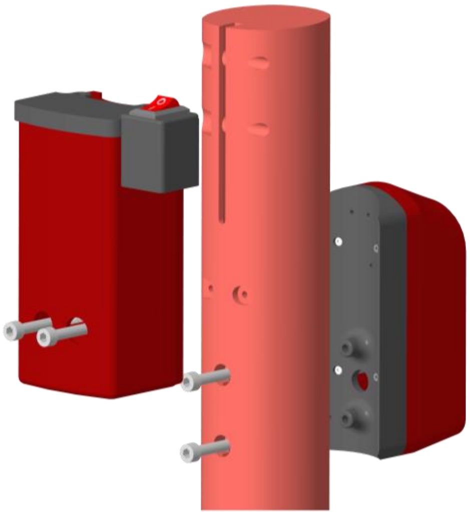


Figure 9 Mounting the modules on the stem.

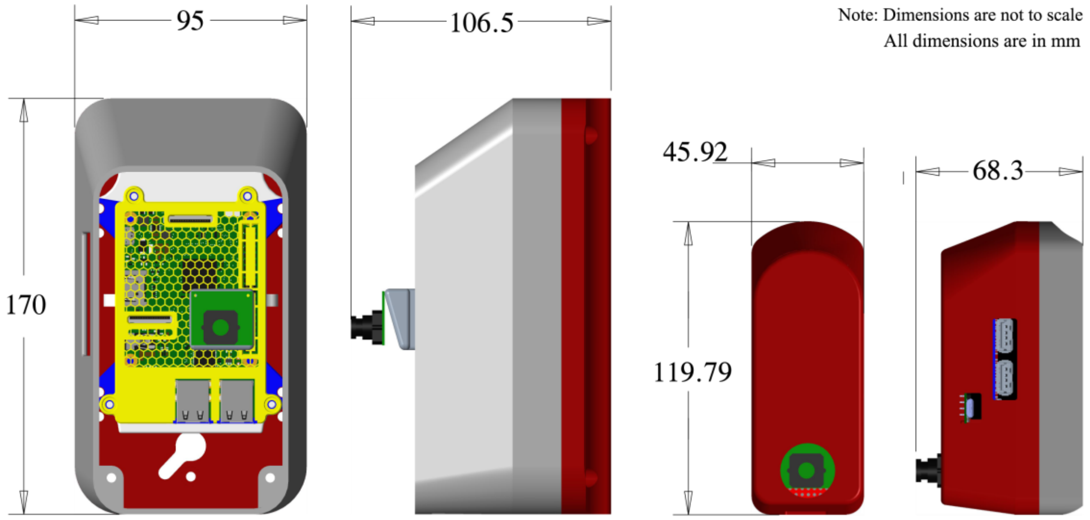


Figure 10 Comparison of dimensions of logger module (Prototype-0 (left) and Prototype-1 (right)).

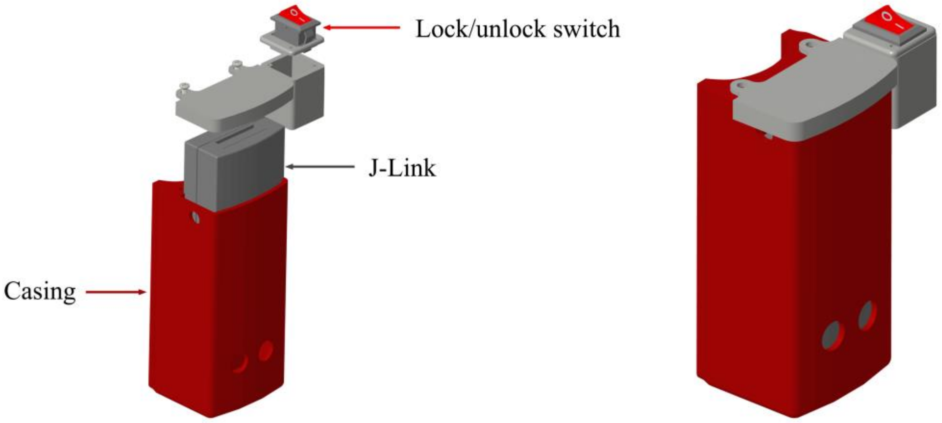


Figure 11 J-Link and power bank module with the lock/unlock switch.

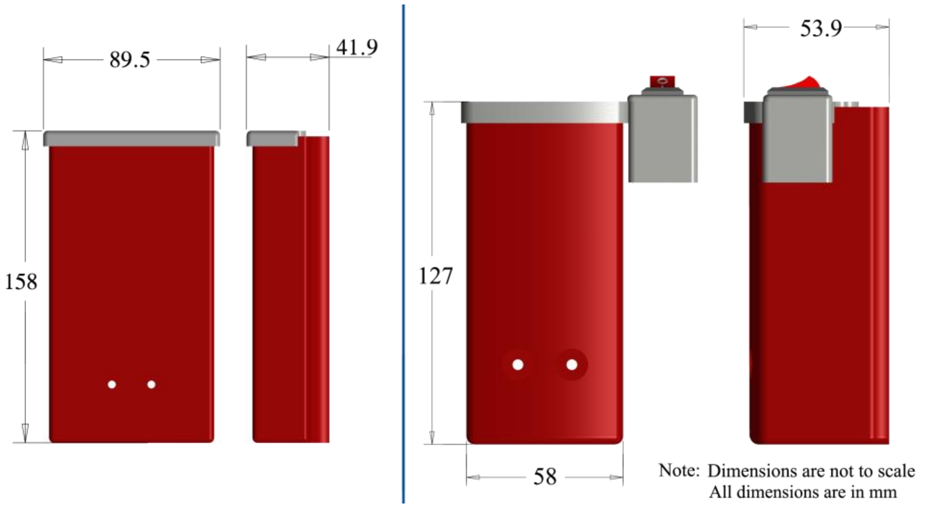


Figure 12 Dimensions of the power bank module from Prototype-0 and J-link module from the Prototype-1.

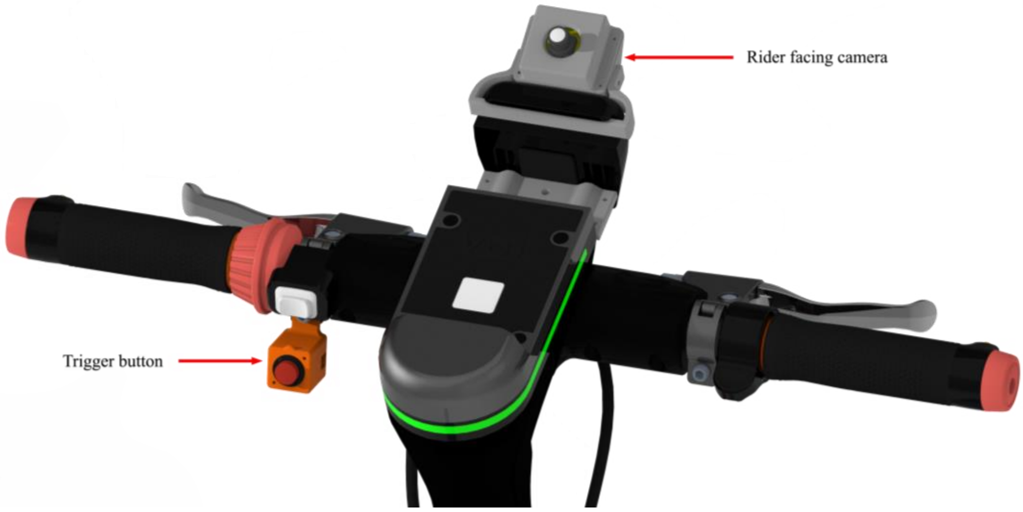


Figure 13 Rider facing camera mount and the trigger button on Prototype-1

2.1.1.3 Voltage converter

One of the requirements of the hardware has been to power the data logger using the battery onboard the e-scooter. The Raspberry Pi has a 5V requirement while the e-scooter is equipped with only 36V (nominal voltage) and 3.3V lines. This resulted in two possibilities to obtain the required voltage level i.e., up-conversion from 3.3V to 5V or down-conversion from 36 - 42V to 5V. The former possibility has been tested initially as it is effective both in terms of packaging and cost. A Polulu U1V10F3 [54] step-up voltage converter has been tested to convert the 3.3V to a 5V supply. However, this approach failed due to the limitation of the current on the 3.3V line on the e-scooter. Thus, the latter approach of down-conversion has been implemented. Given there is a significant fluctuation in the voltage due to regenerative braking and instantaneous accelerations, a step-down converter with a wide voltage range has to be utilized. Traco power TEL 10-4811 [55] with an input voltage range of 36 - 75V and 5.1V, 2A output has been used in the setup. This has been used to power the data logger with the power flow and connections as shown in Figure 14. The issues caused by voltage fluctuation in the logging process have been explained further in the data collection section.

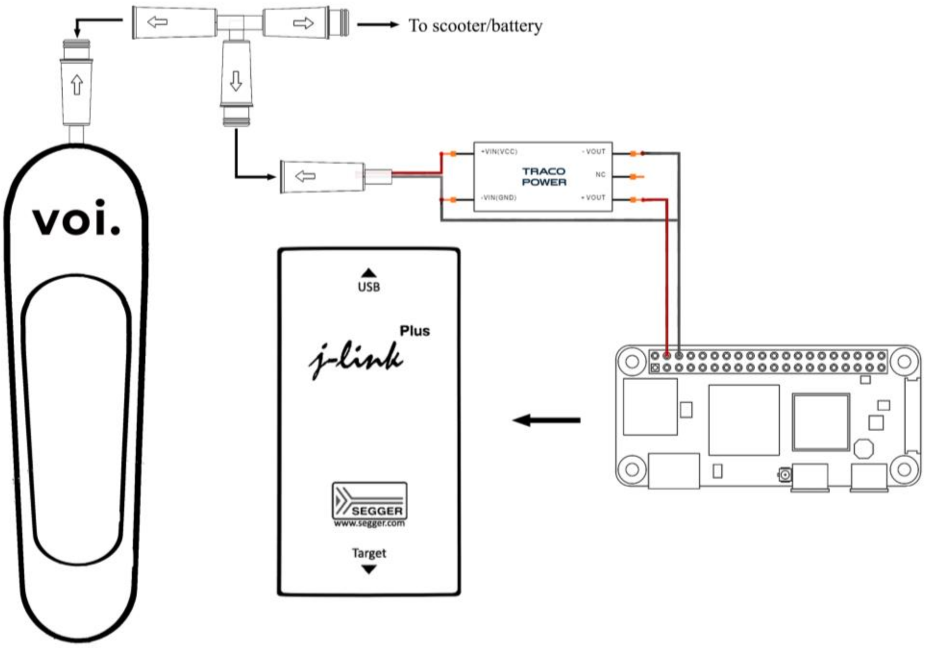


Figure 14 Power flow from the battery to the data logger module.

2.1.1.4 Cameras

The road-facing camera module from Prototype-0 has been retained as it meets all the requirements for the logger. The Zerocam is based on the 5MP OmniVision OV5647 sensor [56] coupled with an M12 220° Field of View (FoV) lens [57] as shown in Figure 15. The horizontal and vertical FoV of the camera with respect to the e-scooter is shown in Figure 16. This FoV does not match the lens manufacturer's specification as the lens and camera module are assembled separately resulting in the discrepancy. Although a No-IR module such as the Arducam B0035 [58] would provide better results in low-light conditions, it would result in poor pictures during the day and also be bulky as compared to the Zerocam. Given the majority of the rides occur during the day, the Zerocam has been used in the logger with software-based tuning done for videos captured in low-light conditions. This is connected to the Raspberry Pi via a Camera Serial Interface (CSI) cable. The rider-facing camera is implemented to obtain information such as the direction the rider is looking during the ride, involvement in any secondary tasks and to obtain information about the number of riders on board during the riding. Arducam B026101 [59] USB based camera with a 2 MP IMX291 sensor coupled with a 175° FoV has been used in the Prototype-1. The camera modules are shown in Figure 15. Although on paper, the USB camera may have lower specifications as compared to the one used in Prototype-0, has a better performance.

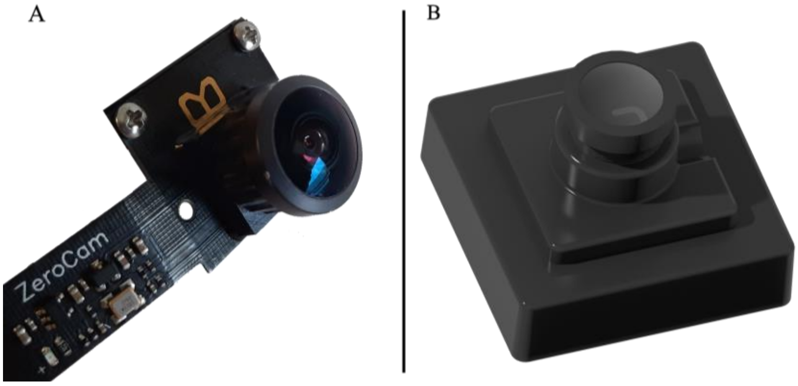


Figure 15 A) Zerocam with 220° FoV lens and B) USB based rider facing camera

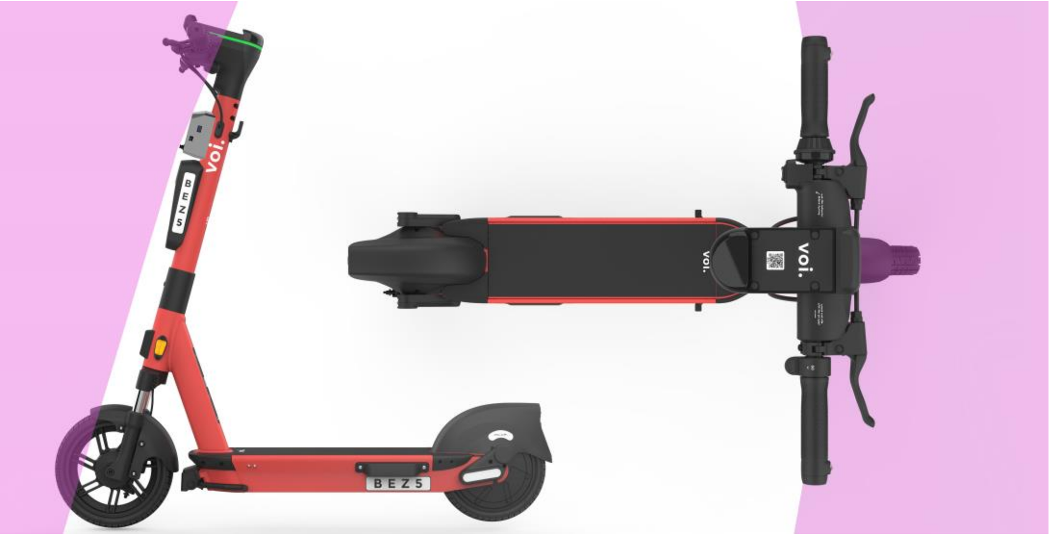


Figure 16 FoV of the road-facing camera

2.1.1.5 Buttons and Switches

A switch as shown in Figure 11 is used to lock/unlock the e-scooter. The Prototype-0 used a push-button for this operation, which resulted in false-positive triggers proving the button to be unreliable. Hence, a switch was implemented with it being ON to keep the e-scooter unlocked and the button being OFF to keep the e-scooter locked. A push-button is mounted on the handlebar below the turn indicator to flag the critical or any noteworthy incidents as shown in Figure 13. The connections of the switch and the push-button with the Pi Zero are shown in Figure 17. 1kΩ pull-down resistors are added to both the switch and the button to prevent any damage to the pins on the Pi.

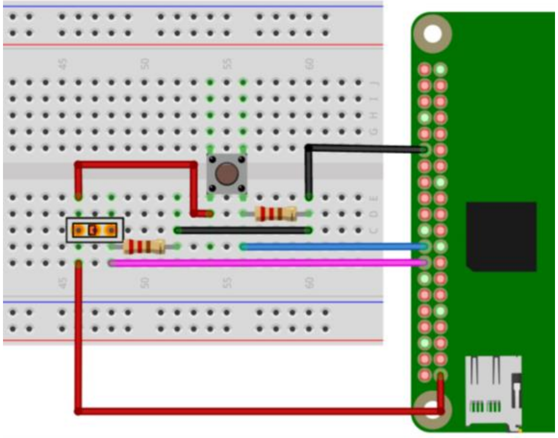


Figure 17 Electrical connections of switch and button in Prototype-1.

2.1.2 Prototype-2

The major change in the 2nd iteration/prototype is the removal of the J-Link debugger and the establishment of a communication protocol. This has enabled further a better packaging of the setup. A custom Printed Circuit Board (PCB) has been developed which, eliminates drawbacks such as loose connections associated with usage of wires. A Universal Asynchronous Receiver/Transmitter (UART) protocol has been implemented for the communication which will be described in detail in section 2.2.2.1. The camera modules used in Prototype-2 is the same one as in the Prototype-1 and is described in section 2.1.1.4.

2.1.2.1 Printed Circuit Board

The major difference to the overall hardware is the elimination of the module on the back of the stem. The removal of J-Link also results in the requirement of only a single USB port for the rider-facing camera. This facilitates better packaging of the logger module by avoiding the use of a USB hub. Replacement of the USB hub has enabled connecting a custom PCB to the Raspberry Pi. The schematics of the PCB as shown in Figure 18.

Figure 19 and Figure 20 show the top and bottom view of the PCB respectively. The voltage converter as described in section 2.1.1.3 is mounted on the top side while the Red Green Blue (RGB) Light Emitting Diode (LED) is mounted on the back. The LED indicates different operations taking place and will be described in the following sections. The 5-Pin connector CN-Main is used to connect the logger to the UART bus of the scooter using the connector similar to the one shown in Figure 14. In addition to this 5-Pin connector, the IoT is equipped with an Auxiliary 8-Pin connector as shown in Figure 21. If the communication protocol in the subsequent setups is modified to I2C or SPI, the logger can easily be adapted by connecting the 8-Pin connector from the IoT to either CN-I2C or CN-SPI connectors on the PCB. The solder pads BTN-3V3 and BTN-DI are used to solder the connections for the trigger button which in Prototype-1 has been soldered directly on the GPIO pins of the Raspberry Pi.

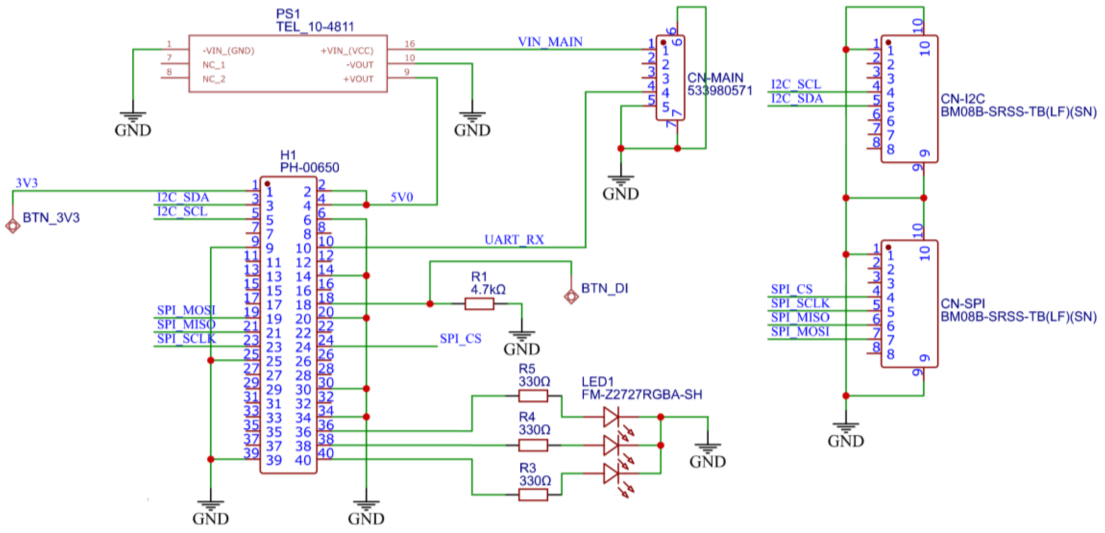


Figure 18 Schematics of the PCB

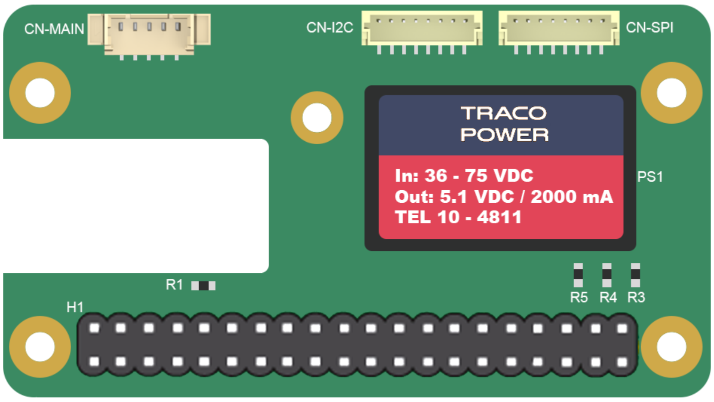


Figure 19 Top View of the PCB

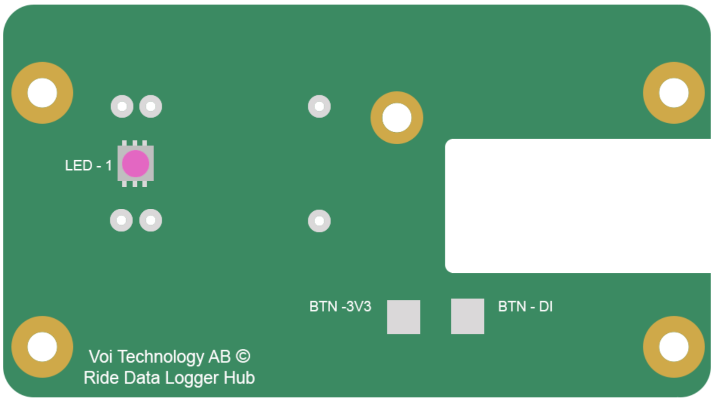


Figure 20 Bottom View of the PCB

Figure 21 IoT of the Voi e-scooter

2.1.2.2 Hardware Casing

The arrangement of each of the components in the data logger module is shown in Figure 22. The PCB is stacked on the Pi Zero 2W with spacers on all four corners to support it structurally. The metal inserts are added to the baseplate which aids in the securing the electronic components in place. The camera is mounted on top of the PCB means of a camera mount. With all components in place, the outer casing is placed on top and screwed from the bottom of the baseplate.

The addition of PCB has enabled mounting the voltage converter in between the Pi and the PCB as compared to the one on Prototype-1 as seen in Figure 23. Additionally, the changes made in the mounting position of the camera module has enabled developed of a compact module. The complete setup in comparison with the logger module of Prototype-1 is shown in Figure 23 and Figure 24.

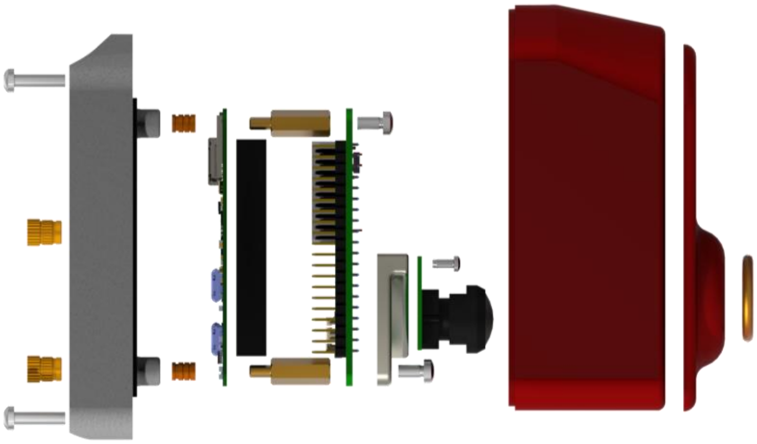


Figure 22 Exploded view of the Prototype-2 data logger module.

The Prototype-2 has an overall dimension of 94.8 x 45.7 x 62.8 mm as compared to the 119.8 x 45.9 x 68.3 mm of the Prototype-1. While the height of the Prototype-2 logger can be decreased further, this will result in the IoT of the scooter blocking the view of the camera. The top casing is split into two pieces, this is to ease the process of manufacturing. An o-ring is used to seal the gap between the lens and the casing. The hole for the LED is covered with transparent silicone to ensure the transparency. These changes are done to ensure that the overall setup is weatherproof. The build of materials essential for the development of the prototype is described in Appendix C.

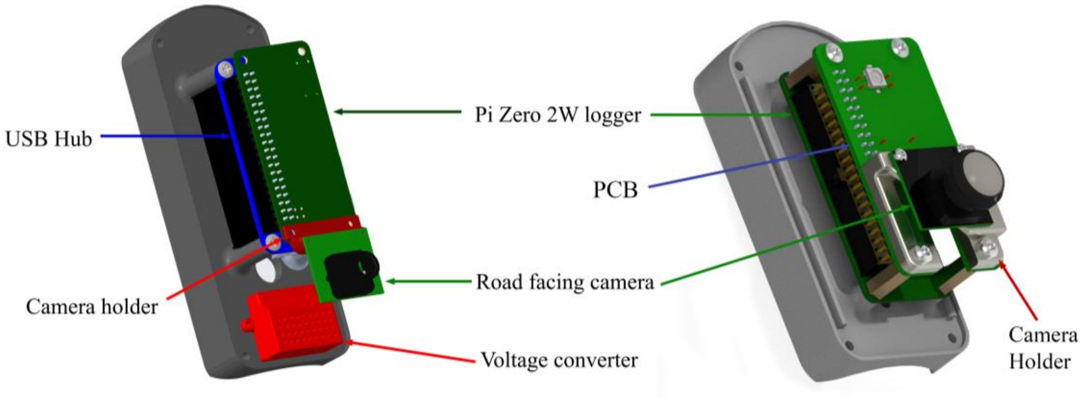


Figure 23 Comparison of component assembly of logger module (Prototype-1 (left) and Prototype-2 (right))

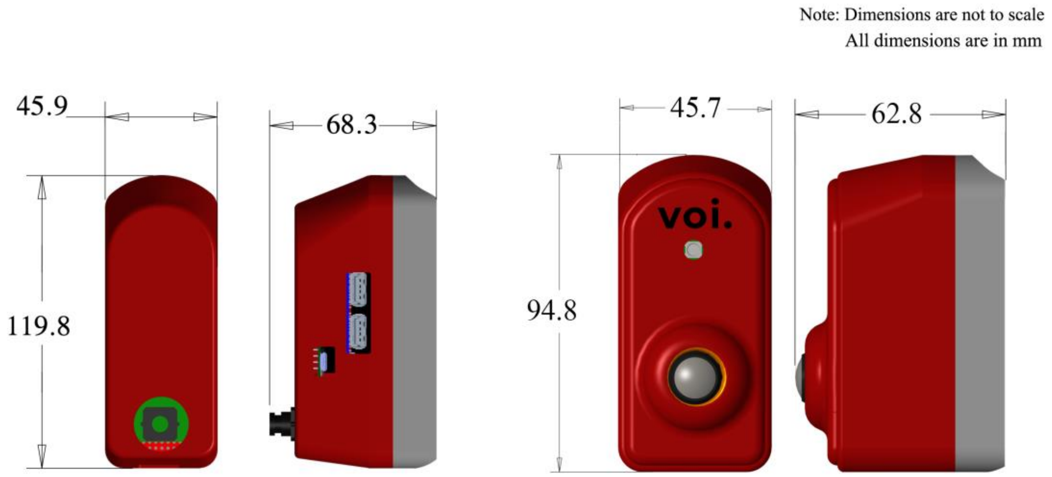


Figure 24 Comparison of dimensions of logger module (Prototype-1 (left) and Prototype-2 (right))

2.1.2.3 Buttons

Given that Prototype-2 involved communication with the e-scooter, the button used to lock/unlock the e-scooter has been removed resulting in a single push button mounted below the turn indicator as shown in Figure 13. Similar to the one in Prototype-1, this button is used to flag any situations that the rider considers uncomfortable or unsafe. The wires are soldered onto the solder pads as described in section 2.1.2.1 The button is also programmed to be used as a reset switch for the logger button press of 10 seconds or longer will reboot the logger.

2.2 Software implementation

2.2.1 Prototype-1

The general data flow for the Prototype-1 setup is shown in Figure 25. The IoT of the e-scooter is equipped with an IMU, Gyroscope, and GPS which provides details of the accelerations, inclination and the global position of the e-scooter respectively. The motor controller provides the battery voltage and battery current which is also stored in IoT. For the Prototype-1 a J-Link debug probe is connected to the IoT which then can be connected to the Pi through USB. The data acquisition system program is developed to record the kinematic data and the cameras simultaneously. Multithreading [60] is used to run different operations concomitantly.

In order to eliminate the need for a person to manually collect the data from each e-scooter, the process is automated with data being uploaded if and when an internet connection is available. Data upstream management is developed to automate the data uploading and management. An overview of how each of these programs works on the Pi is shown in Figure 26.

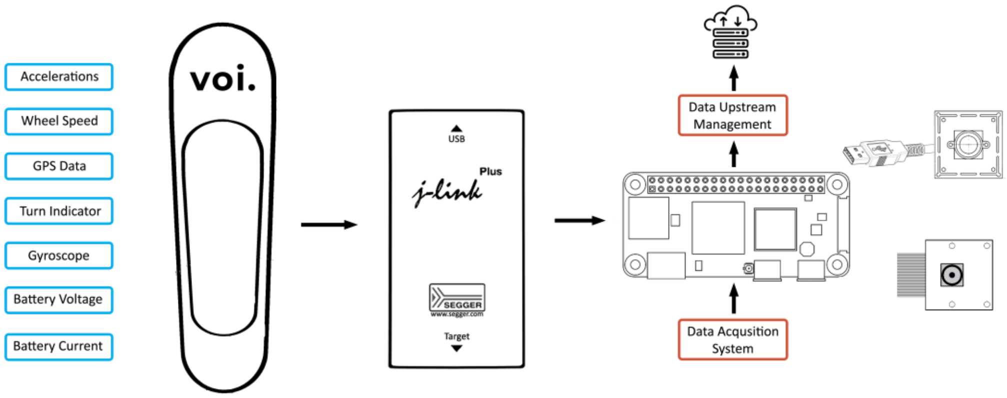


Figure 25 Data flow in the Prototype-1.

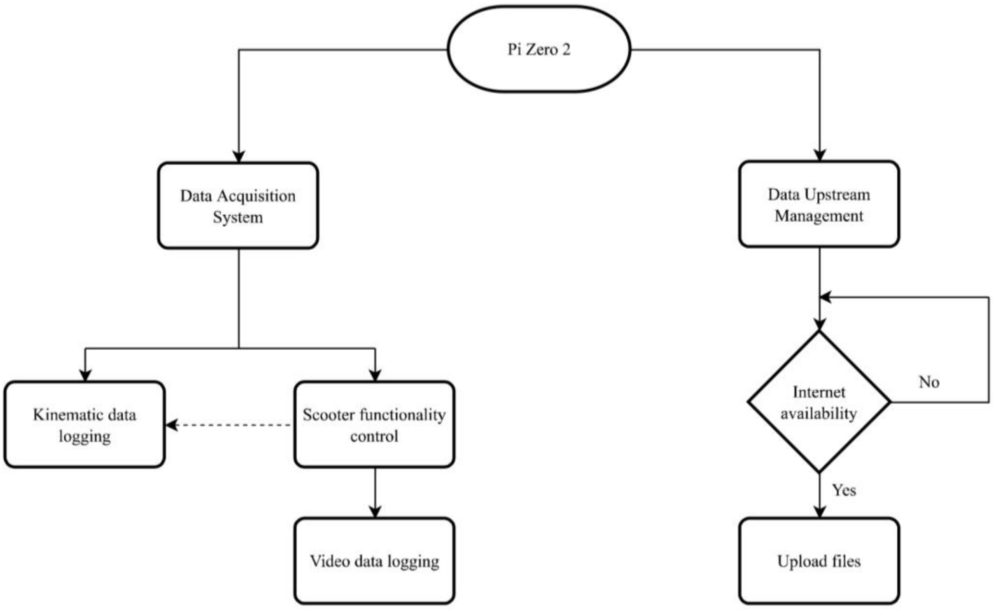


Figure 26 Overview of programs running on Pi in Prototype-1.

2.2.1.1 Scooter functionality control

Given that Prototype-1 majorly involved hardware optimization, the data acquisition system program works similar to the one developed as a project by Schmidt et al. [47]. Modifications are made to account for the change in the button used to lock/unlock. The program runs in an infinite loop that constantly tracks the state of the lock/unlock switch. Figure 27 shows the flowchart of how the program manages the locking/unlocking of the e-scooter. The unlock operation simultaneously starts the kinematic and video data recording and the locking of the e-scooter is responsible to terminate the recording of data. This is illustrated by the connector blocks 1 and 2 which in Figure 28 and Figure 29 indicate the effect of e-scooter functionality on data logging.

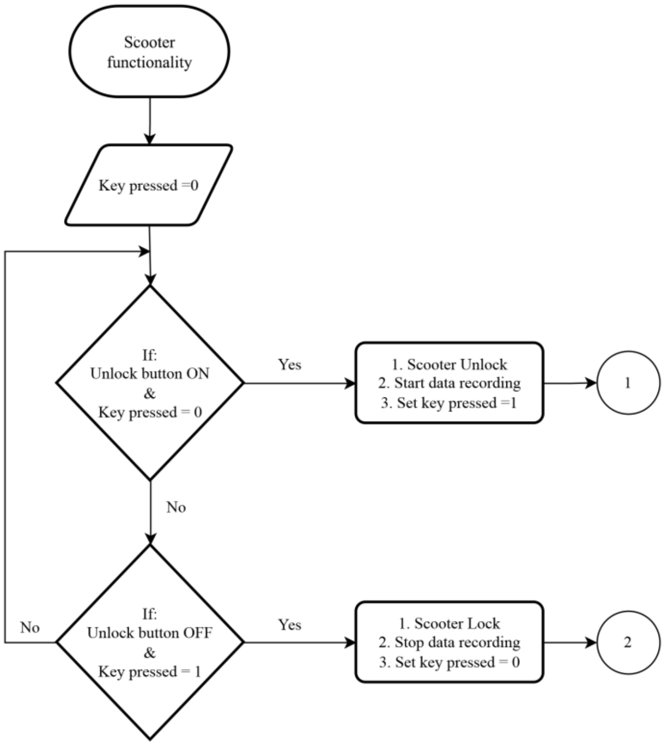


Figure 27 Flowchart of the program controlling the e-scooter functionality in Prototype-1.

2.2.1.2 Video data logging

As described in section 2.2.1.1 the video data logging is controlled by means of the button controlling the e-scooter functionality. An important variation between the two cameras is that the road-facing camera uses the PiCamera library while the rider-facing camera uses the FFMPEG library. The timestamp is annotated on the top of the video to facilitate syncing of the video. When the trigger button is pressed to mark events, the background of the time annotation changes to red for the duration the button is pressed. This is to ensure easy identification of the events and achieve sync between the kinematic and video data. The road-facing camera records 30 frames per second (FPS) with a resolution of 720x532p. The rider-facing camera records in a resolution of 640x480p with an average of 30 FPS. '.h264' format has been used for recording the road-facing camera while '.avi' format has been used for recording the rider-facing camera. Given the bright days during the spring and summer, the brightness and contrast level for the road-facing camera remains unmodified. The brightness is increased to 65 (default value is 50 with a range of 0 to 100) and the contrast is decreased to -0.1 (default is 0 with a range of -1 to 1) for videos recorded between dusk and dawn to compensate for the reduced lighting conditions. The complete process of video data logging is shown in Figure 28.

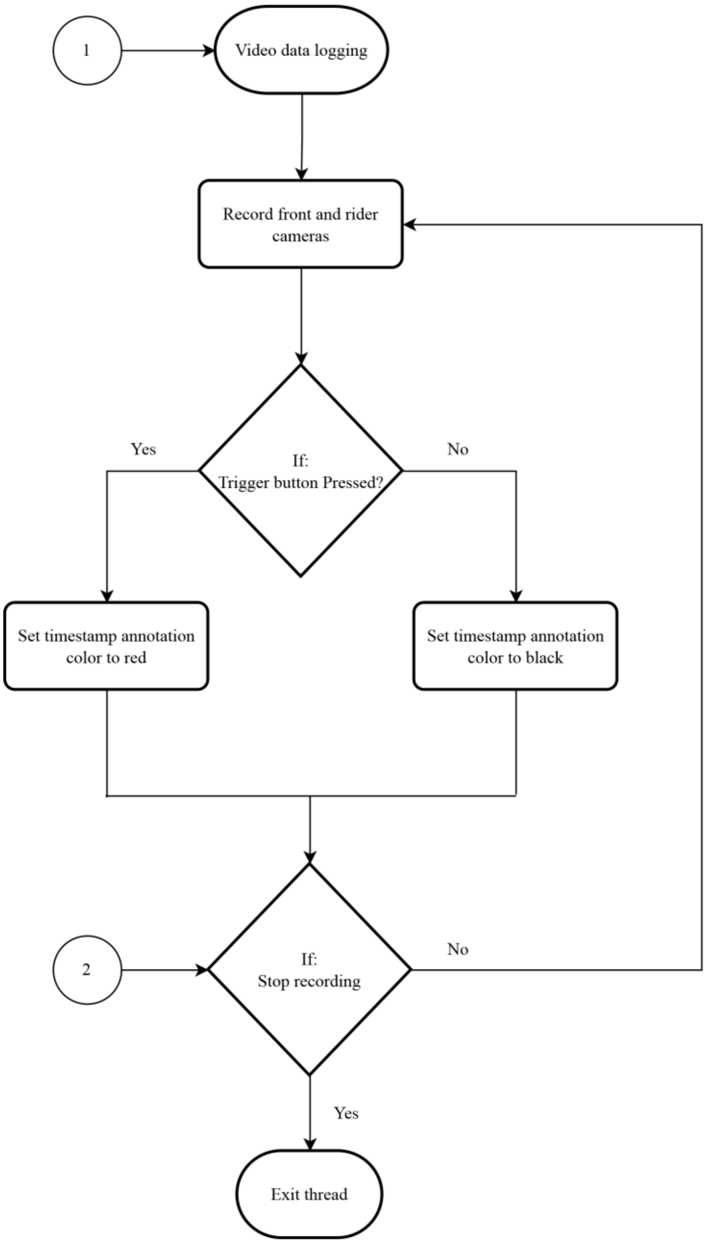


Figure 28 Flowchart of the video data logging thread for Prototype-1.

2.2.1.3 Kinematic data logging

Unlike the camera thread that runs only during the recording, the kinematic data logging thread runs in an infinite loop. This is to avoid data loss that can occur during the initialization of the J-link communication. The J-link debug probe sends out all the information that it receives from the IoT. Only the lines starting with 'DATALOG:' contains relevant information and thus when detected is extracted and labelled with a timestamp in addition to the trigger flag. This is saved in a '.csv' format with one file per ride. Figure 29 shows the flowchart of the kinematic data logging on the data logger. Since the same timestamp is used to mark the dataset and the video the sync is achieved during the process of recording. The average frequency of data reception is 10 Hz. But on a few occasions, the UART bus connecting the e-scooter and IoT gets blocked resulting in missing data points.

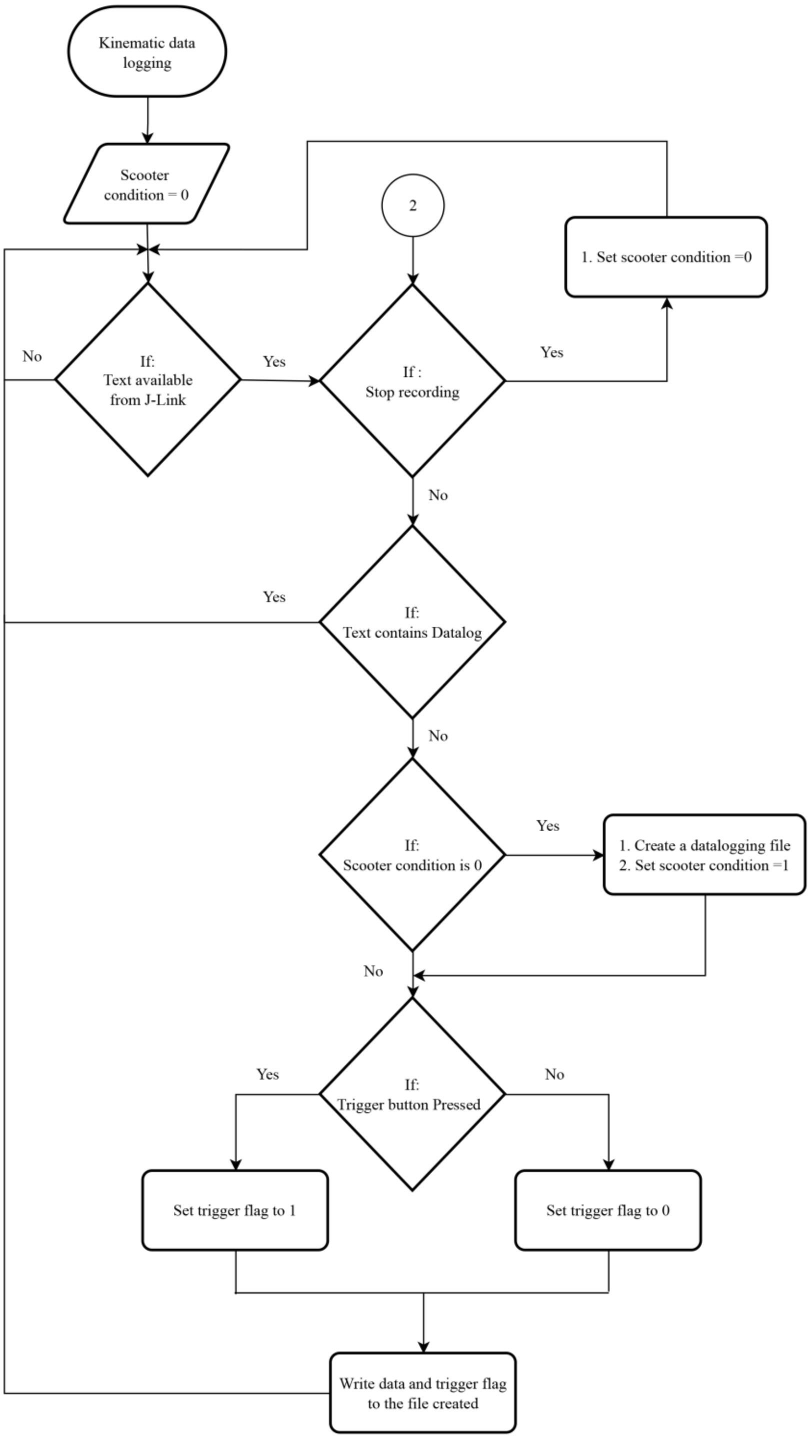


Figure 29 Flowchart of the kinematic data logging thread for Prototype-1.

2.2.2 Prototype-2

The general data flow for the Prototype-2 setup is shown in Figure 30. The major update from the previous generation prototype is the establishment of communication directly with the e-scooter resulting in a more robust, secure setup. The IoT uses UART to communicate with the e-scooter. The same framework is used for communication with the Raspberry Pi. The data is stored in encrypted container which is unlocked during the boot. The overall process carried out by the logger is shown in Figure 31. Unlike the Prototype-1, the Pi Zero 2 based logger in the Prototype-2 is only used for recording video data while the kinematic data is recorded separately on the IoT of the e-scooter.

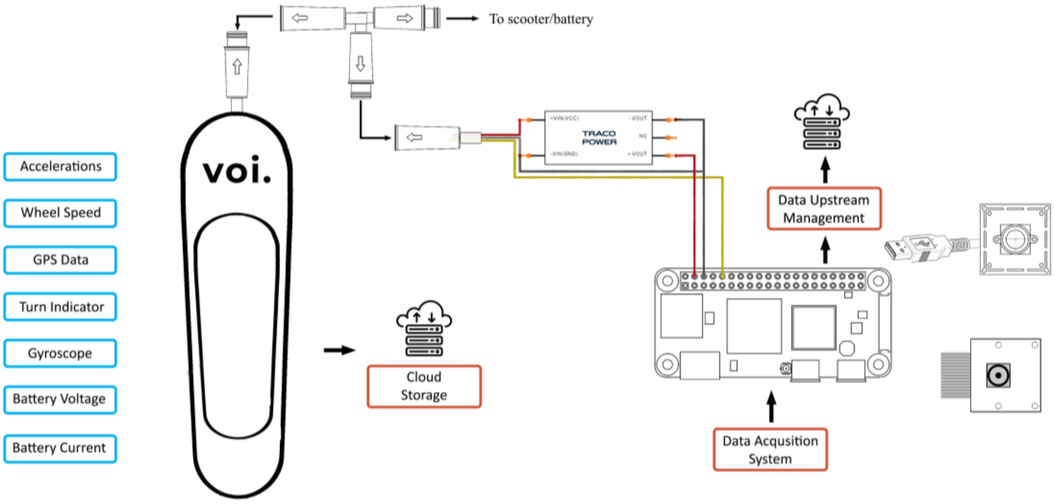


Figure 30 Data flow in the Prototype-2.

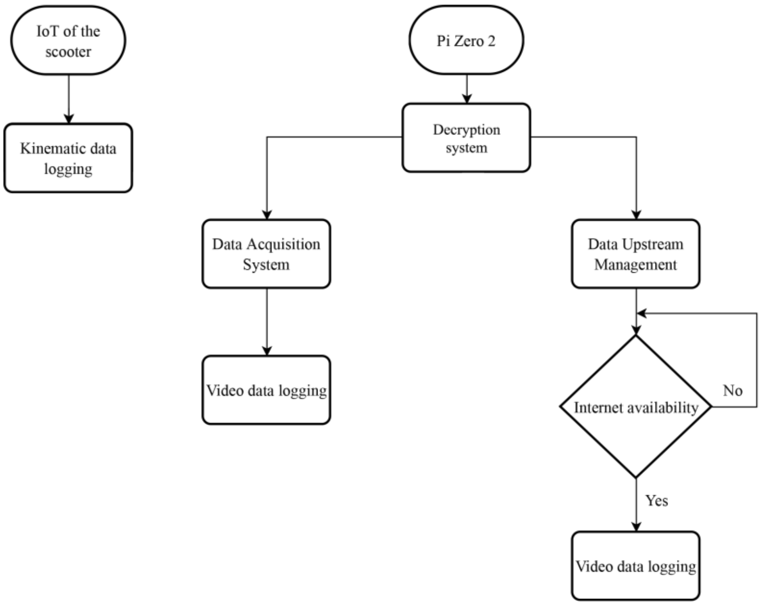


Figure 31 Overview of data acquisition and management in Prototype-2.

2.2.2.1 Communication Protocol setup

The UART bus is split using a Y cable to add a second receiver i.e., the data logger as shown in Figure 30. The IoT sends messages to indicate the locking and unlocking of the e-scooter. The custom message is developed and transmitted by the IoT for a span of 400 ms during each of the operations. Given the message has a different length contrary to the regular messages, the e-scooter would ignore it completely. Thus, the regular communication between the IoT and e-scooter remains unaffected. This enables the e-scooter to be used as a regular e-scooter. Initially, the possibility for a two-way communication has been tested but resulted in the communication between the e-scooter and the IoT being disturbed and subsequent stoppage of the e-scooter itself.

2.2.2.2 Data encryption

A partition is created on the SD card of the logger. The data logging software and the video data collected is stored in the container which is encrypted using Linux Unified Key Setup (LUKS) [61]. This helps preventing unauthorized access to the data in case of theft and in addition comply with the General Data Protection Regulations (GDPR) [62]. During boot, the container is decrypted, and data is saved in the container during logging operation.

2.2.2.3 Video data logging

The video logging works similar to the one in the Prototype-1. The main thread constantly keeps a check for the messages on the UART bus. During unlock and locking of the e-scooter, the IoT sends a custom message which is identified by the logger based on the length of the message. Given the logger is not connected to the internet when on the streets resulting in the clock of the raspberry pi not being accurate, the message from the IoT includes the Coordinated Universal Time (UTC) timestamp which is used to update the clock on the logger. The updating of the clock becomes essential when the logger is left with no internet access for a long duration as it tends to lose count of time. The updated time ensures the sync between the data and the videos. Based on the current state of the e-scooter as received in the UART signal and the previous state of the e-scooter as stored in a variable with the program, either the video recording is started or stopped. The complete process of video data logging is shown in Figure 32. As the e-scooter changes its state from unlocked to locked, the storage available on the data logger is checked and if found to be below a threshold of 5 Gigabytes, the logger is turned off. This is done to avoid disturbing the normal functionality of Raspberry Pi. The RGB LED is programmed to glow green during the video data recording operation.

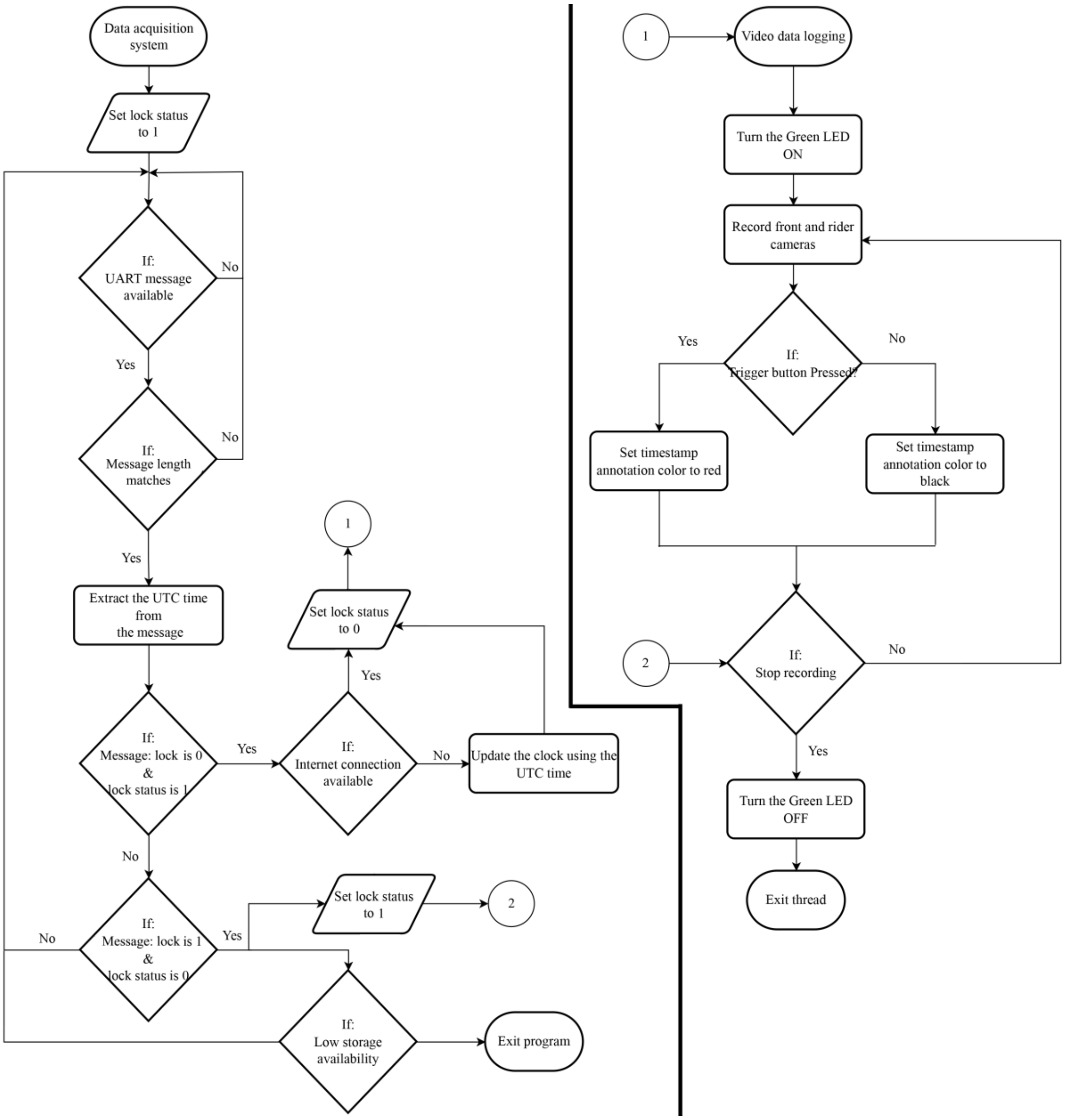


Figure 32 Flowchart of the video data logging for Prototype-2.

2.2.2.4 Kinematic data logging

The kinematic data is logged on the IoT of the e-scooter. The IoT being connected to the internet can upload the log files as soon as it becomes available. Every time a user ends the ride, the file for that ride is then uploaded to the google cloud service. This reduces the possibility of data loss due to vandalism or any other technical issues. The data is logged at 10 Hz which includes parameters such as accelerations, tilt angles, GPS coordinates, wheel speed, battery voltage and current. Although it would be ideal to have the video logging files uploaded simultaneously, the lack of internet connectivity of the data logger eliminates the possibility. Also, the videos have a relatively large file size which will result in a longer upload duration making the process non-feasible.

2.2.3 Data upstream management

The major update from Prototype-0 to Prototype-1 is the elimination of manual extraction of logged data. Automating the process helps save cost and time, making the process streamlined. Rclone [63] is used as the tool to upload the data to Google drive. The program runs as shown in Figure 33. The detailed program can be found in Appendix A.

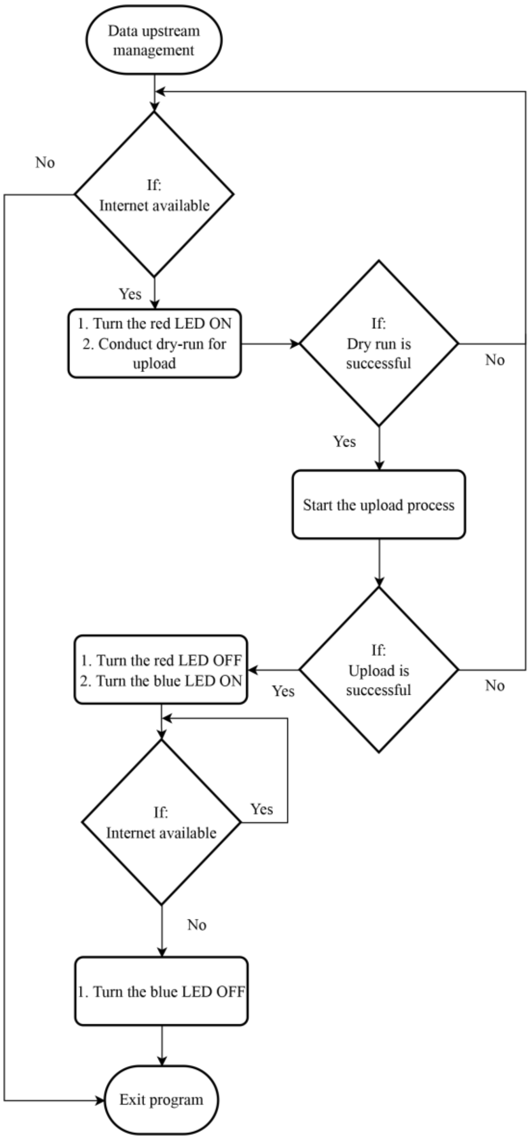


Figure 33 Data upstream management.

On starting the program, the internet availability is checked, and if no connection is obtained, the program is stopped. If internet access is confirmed the red LED is turned ON to indicate the initialization of the upload process. Although upload can be started directly, since the files are being moved, Rclone recommends a dry-run which simulates the upload and enables detection of any potential issues. If the dry-run is successful, the program proceeds to the actual upload process. In either of these steps results in an error, the program is reset and repeats the aforementioned steps. Upon successful upload, the red LED is turned OFF and the blue is turned ON. The blue LED stays on until the logger loses internet connectivity. The Table 1 lists the color of the LED and the corresponding operation taking place on the logger. The script is run at regular intervals which will be described in detail in the next subsection. To avoid the script running when an upload has already been successful the script is not terminated until the logger loses internet connection. The same system is employed in the Prototype-2 for uploading the video data from the logger.

Table 1 LED colour and corresponding operation

LED Color	Operation
Green	Recording
Red	Uploading
Blue	Upload complete

2.2.4 Scheduling the program

The data acquisition system should be running all the time the e-scooter has a power supply. To enable this the script starts every time the logger is rebooted. This is done by setting up a Cron job [64] on the logger. The reboot can occur either by holding down the trigger button for 10 seconds or during the battery swap operation of the e-scooter. To avoid a forceful reboot during battery swap, the Pi shuts down as soon as the battery hatch/lid of the e-scooter is opened and then boots up when the new battery is plugged in. The data upstream management on the other hand is run every 10th minute. Given the script only runs when connected to the internet, even though it is started every 10th minute it does not consume any additional computational power. When the upload is in progress, the duration of the upload can be longer than 10 minutes, to avoid multiple upload operations the file is locked. Flock [65] helps lock the file and thereby makes it unavailable to run multiple times simultaneously. Thus, a flock based Cron job is set up to run every 10th minute.

2.3 Data Visualisation Toolbox

To enable the analysis of the data being collected, it is essential to have a Graphical User Interface (GUI) to facilitate the process. The GUI developed by Schmidt et al. [47] is based on MATLAB. While the GUI has been a very helpful tool, there are several drawbacks. This includes the inability to play both the videos simultaneously and have seeking (forwarding and rewinding) for the videos. In addition, MATLAB is a paid application and computationally intensive. Hence, a Python-based GUI has been developed as a part of this thesis. Figure 34 shows the GUI during a playback. The overall layout can be divided into 10 parts.

The user can select the dataset to be analyzed by clicking the load button. This opens a popup allowing the user to select the kinematic data file for that ride. The program then finds the road-facing and the rider-facing video data for the selected dataset. The second panel includes the list of variables recorded during the ride and allows the user of the GUI to select the variable to be visualized. Details of each of the variables are further explained in section 3.4. Once the user selects the dataset, the graph with the values on the Y-axis and timestamp on the X-axis is plotted. The variable selected by default shows the accelerometer reading in the x-direction. The video data plays simultaneously in the 6th part of the layout. It is important to have a marking on the graph indicating what part of the ride is currently being played on the video frame. This will help understand the kinematics while simultaneously getting a visual representation of the rider and surroundings. The annotation 4a in Figure 35 shows how this marking appears to the user of the GUI. As explained in section 2.1.1.5 the rider has the provision to mark any specific events by means of the trigger button. Any such triggers are marked on the graph as illustrated by annotation 4c in Figure 35. The yellow box starts 5 seconds before the trigger span till 5 seconds after the trigger. If multiple triggers are detected within the span of a second, they are mapped to be a single trigger event. The user also can zoom in on the graph using the slider marked by annotation 4b in Figure 35. The zoom feature helps understand the finer details over a short interval of time. The blinker status of the e-scooter is visualized in the 3rd part of the layout as shown in Figure 34 and functions based on the playback of the videos. The GPS coordinates are plotted on the graph in part 5 of the GUI. Similar to the kinematic data graph, the playback of the video is marked with the annotation 5b as shown in Figure 36 with trigger indicators plotted on the graph as shown by annotation 5a. The GPS plot is based on the OpenStreetMap [66].

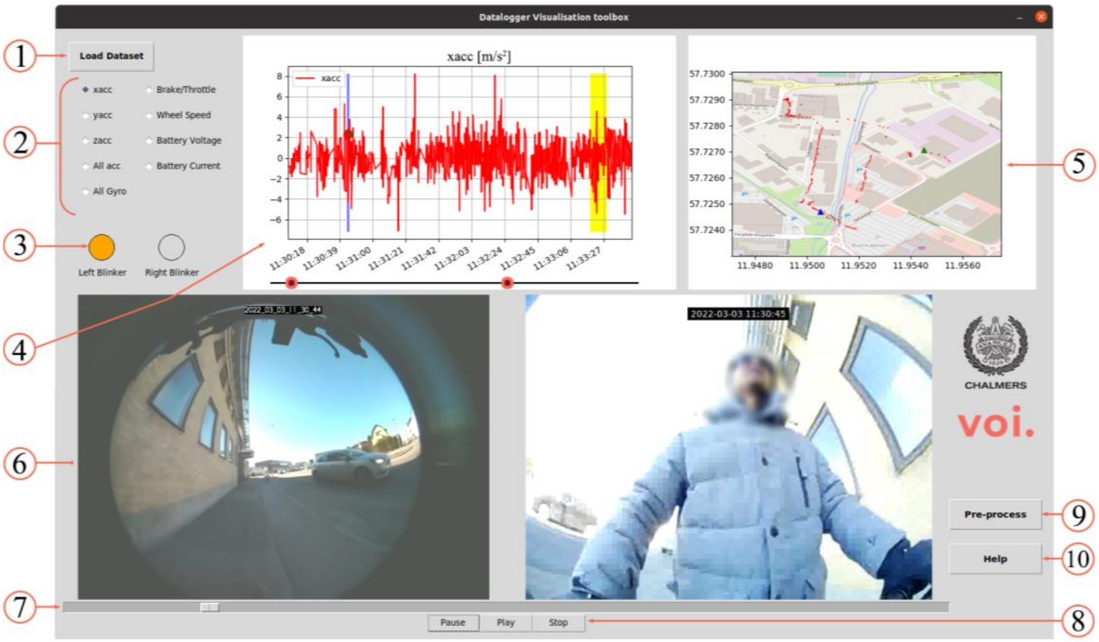


Figure 34 Python based GUI developed for data analysis

The user of this tool can forward to a specific part of the video, pause, play and stop it using the controls provided in parts 7 and 8. The raw data collected requires to be pre-processed before the visualization or analysis is carried out. On clicking the pre-process button, the GUI allows the user to select the folder containing the datasets. The processes involved in this step are explained further in detail in section 4. The help button that forms the part 10 of the GUI provides detailed insight onto the toolbox.

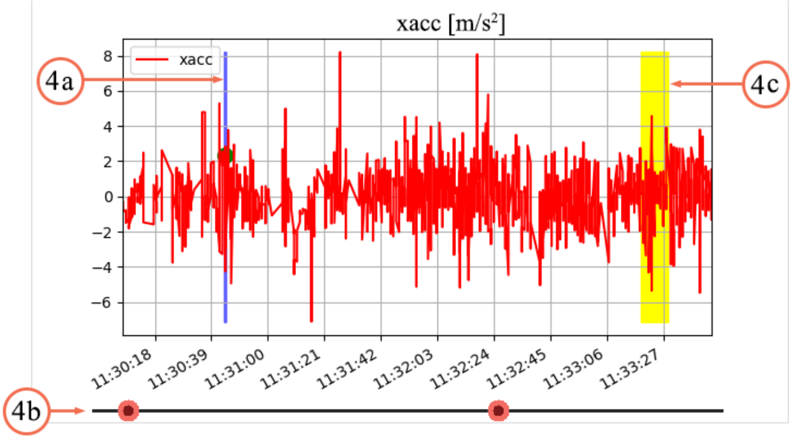


Figure 35 Features on the kinematic data visualisation graph

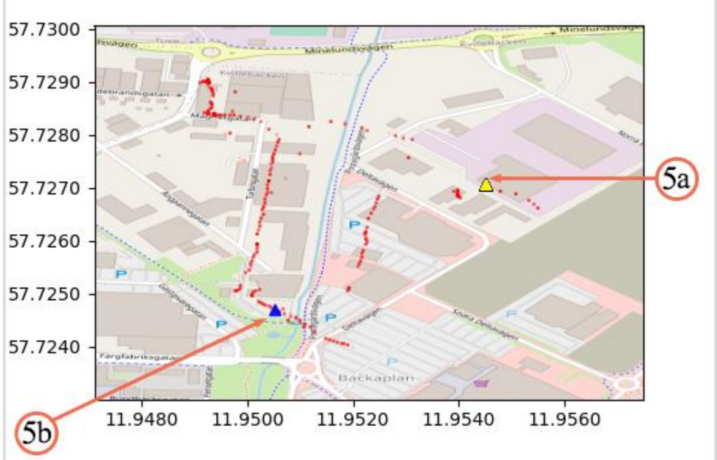


Figure 36 GPS co-ordinates of the ride plotted on the map

3 Data Collection

The objective of data collection is for testing the logger's performance and further analyze the quality of the logged data. The data collection also directs the modifications required that enable development of a robust and reliable data logger for a following larger naturalistic riding study.

3.1 Pilot testing

The pilot testing involved the identification of potential issues during operation. This has been carried out throughout the development stages of the prototypes. Several road surfaces, lighting conditions, riding style, and ride duration have been tested. The e-scooter has been ridden on road surfaces such as cobblestones to identify any possible issues caused due to rapid vibrations of the e-scooter frame. The varying lighting condition at the onset of spring enabled testing of the adaptability of the camera module. Various other tests such as rapid accelerations, jerky riding, stop-and-go rides within the city and harsh braking have also been carried out. The ride durations starting from as low as a minute to an hour-long have been tested to identify the performance of the logger and test any possible frame drops etc. during the prolonged data logging.

The Prototype-1 in the initial testing failed, resulting in the loss of data. As mentioned in section 2.1.1.3, this has been identified to be caused due to voltage fluctuation due to rapid acceleration and regenerative braking. The J-link being sensitive to voltage supply shut down with every voltage drop resulting in failed logging. To eliminate this, an external power supply had to be added to eliminate the fluctuations. But given that Prototype-2 would eliminate the J-link debugger, the issue has been given a low priority. The mounting point on the casing used in Prototype-1 also has been identified to fail under extreme vibrations. Although the e-scooter being exposed to these operating conditions is unlikely, the casing has been modified with the addition of reinforcement at the fracture point.

Prototype-2 involved several tests to verify the communication protocol. Modifications have been made with each test. This includes testing the different lengths of the message being transmitted along with the parameters that are to be included in the message. Tests have been conducted to identify the time intervals each of the messages must be transmitted by the IoT. Effective communication has been identified to be the transmission of messages in short bursts during the locking and unlocking operation. The message for operational errors of the e-scooter is to be transmitted till the error has been resolved. This enables taking appropriate actions such as shutting down the logger.

3.2 Participant Selection

Given the logger is a prototype, when selecting the participants only those that had the possibility of storing the e-scooter in a secured place have been selected. In addition to this, the participants are required to not leave the e-scooter unattended in a public place given the chances of a non-participant riding it. The risk of vandalism and theft has also been a matter of concern for deciding not to leave e-scooter in public areas. These criteria are known to result in a bias in the data collected but are essential during the prototype stages. A total of four participants took part in the data collection. Some background information about the participants is shown in Table 2. Each participant has the e-scooter with them for a week. The participants have been asked to mark any uncomfortable or unsafe situations using the trigger button. The participants have also been asked to check for the status LED to be glowing green at the beginning of the ride to confirm that the logger is working as intended.

Table 2 Background information of the participants

	Average	Standard deviation	Minimum	Maximum
Age (years)	35.5	8.96	24	48
Height (centimeters)	178.75	4.21	174	185

3.3 Subjective data collection

The subjective data has been collected to understand more about the participant and their experience with e-scooters. This will enable comparison and help map various individual characteristics with the riding behaviors. The data has been collected in the form of a questionnaire. The participants have been requested to fill in the form after having taken part in the study. The questions have been formatted to include Yes/No, multiple-choice, Likert scales and open-ended questions. The questionnaire has been divided into three sections with the first being the demographics followed by the usage of the e-scooters and finally their opinions about e-scooters in general as seen in Appendix B. The demographics involve the rider’s age, height, gender, employment and the mode of transport regularly used. The subsequent section includes questions such as ownership of an e-scooter, average e-scooter riding duration, previous experience with e-scooters, season-based variation in e-scooter usage etc. The final section includes opinion-based questions such as the safety of the e-scooters from the perspective of the participants, helmet usage on e-scooters etc.

3.4 Objective data collection

The objective data includes the kinematic data and the video data collected by the IoT and the e-scooter respectively. The frames from the videos captured using the road-facing and rider-facing camera are as shown in Figure 37 and Figure 38 respectively.

Figure 37 Frame from the video recorded by the road-facing camera during data collection

Figure 38 Frame from the video recorded by the rider-facing camera during data collection

As seen in Figure 37 and Figure 38 the timestamp is marked on the top of the video. This helps sync the data with the video.

The kinematic data recorded on the IoT of the e-scooter is stored in a '.csv' file which includes several parameters. Detailed information on each of these parameters is listed in Table 3. The coordinate system of the IMU and gyroscope is shown in Figure 39. The stem of the e-scooter is inclined 16.3° to the vertical resulting in the Y and Z axis being rotated by the same angle with respect to the ground/vehicle coordinates.

Table 3 Sensor reading description from the e-scooter

Variable	**Description**	**Unit**	**Range**
t	Uptime of the e-scooter	ms	
UTC	Coordinated universal time	s	
ax	Accelerometer reading in x direction	m/s^2	
ay	Accelerometer reading in y direction	m/s^2	
az	Accelerometer reading in z direction	m/s^2	
rx	Gyroscope reading in x direction	degree/s	
ry	Gyroscope reading in y direction	degree/s	
rz	Gyroscope reading in z direction	degree/s	
v_wheel	Sensor based wheel speed	km/h	Max speed limit of 20 in Sweden
v_GPS	GPS based e-scooter speed	km/h	

lat	Latitude co-ordinates from the GNSS module	degree	
lon	Longitude co-ordinates from the GNSS module	degree	
tot_dist	Distance travelled by the e-scooter	meter	
blink	Turn indicator status		0 - Off 1- Left 2 - Right
throttle	Hall-effect sensors in the throttle		45 - 200
brake_l	Hall-effect sensors in the left brake (Left brake is connected to the rear wheel)		45 - 200
brake_r	Hall-effect sensors in the right brake (Right brake is connected to the front wheel)		45 - 200

Figure 39 Co-ordinate system of the IMU and Gyroscope

Given the project is scaled to include the logger on multiple e-scooters, the chances of two e-scooters being ridden at the same time exists. Thus, the kinematic data file and the video files are named in 'IMEI-Timestamp' format where the timestamp is the UTC the ride was completed and IMEI is the unique identification number for each of the e-scooters.

During the entire data collection process, the video logging failed once, as explained in section 2.1.2.3 the trigger button can also reboot the raspberry pi which in the initial stages had been set to 2.5 seconds. Thus, a trigger operation carried out by the participant resulted in the logger rebooting during the ride. This has been corrected by changing the threshold for the reboot to 10 seconds and power-cycling the pi.

4 Data Analysis

The aim of the data analysis is to assess the quality of the data collected and potential points for improvement before a large fleet of e-scooters is equipped with loggers for naturalistic data collection. However, a detailed data analysis can be conducted on several aspects such as rider behavior analysis, safety-critical situation analysis etc. which will be explained in detail in section 5. The limited timeline of the thesis has resulted in the analysis being constrained to quality assessment and some preliminary tests to showcase the potential of the datasets.

4.1 Pre-processing the data

The raw data collected has to be pre-processed before any analysis can be carried out. This ranges from filtering of the kinematic data to the conversion of the video files to relevant formats.

4.1.1 Filtering kinematic data

As a result of a bug in the kinematic data collection software, some data points in the log file are found to contain irrelevant information. Given the uncertain cause of this issue, these data points are thereby eliminated during the process of filtering. Each of these points has values for the brake lever position and throttle position outside the range described in Table 3. Hence adding these constraints removes any inconsistent data points. There are several instances during which the e-scooter tends to lose the GPS signals resulting in the coordinates being marked as 0.0 N and 0.0 E. When plotting the GPS coordinates on the map as shown in Figure 36, only those data points that include a coordinate within Sweden are considered. This filtering is done only for this plot and does not apply to the entire dataset.

4.1.2 Time delay between the videos and the kinematic data

From the numerous datasets collected a time delay between the start of the road-facing camera and the rider-facing camera has been identified. The magnitude of the delay has been identified to be an average of 2 seconds. Given the rider camera uses an external library FFMPEG, this delay can be attributed to the time taken to create a file and initialize the camera. The delay cannot be eliminated by any modifications and is bound by software and hardware limitation on the logger. Hence this is eliminated in the post-processing by skipping the first 2 seconds of the road-facing camera. Given the assumption that there is no loss in data transfer, the road-facing camera starts recording the same instance the e-scooter is unlocked. The kinematic data on the other hand is observed to have irrelevant data which spans for an average of 10 seconds which is filtered as described in 4.1.1. It has also been noted that an average delay of 2 seconds caused due to creation of a new data file and initialization of the data log on the IoT. Thus, a 12 second delay between the kinematic data and road-facing camera is observed. While it is not ideal to have data loss, it is assumed that the initial 12 seconds will not have any relevant information as the rider needs more time to unmount the e-scooter from the stand and start the ride. Also given the time required for identification and correction of the issue is much more than the value these data points add to the analysis this issue has been ignored. Table 4 provides details of the 3 datasets collected by a participant and the timestamps of the first data points in each of the videos and kinematic data.

Overall, the video from road-facing camera is skipped by a total of 12 seconds while the video from rider-facing camera is skipped by a total of 10 seconds to compensate for the delay in the kinematic dataset. Each of these delays has been set as variables in the GUI thereby any advancement in the future resulting in lower delays can be tuned easily.

Table 4 Timestamps of data collected

Dataset Id	Timestamp of the first available datapoint		
	Road-facing video	Rider-facing video	Kinematic data
2022_04_29_07_22	07:22:47	07:22:49	07:22:59
2022_04_28_18_25	18:25:30	18:25:33	18:25:42
2022_04_27_07_04	07:04:37	07:04:38	07:04:49

4.1.3 E-scooter stem angle compensation

As described in section 3.4 the stem of the e-scooter is at an angle of 16.3 degrees o the vertical. To understand the behavior of the e-scooter it is necessary to base this relative to the ground. This is compensated by the rotation transformation of both the Y-axis and the Z-axis. Equations 1 and 2 provide the transformation operation where $A_y{}'$and $A_z{}'$is the transformed accelerometer readings, A_y and A_z are the sensor readings from the accelerometer in y and z directions respectively and θ is the stem angle. The same applies to gyroscope readings in y and z directions. The axis before and after transformation are shown in Figure 40.

$$A_y{}' = A_y * \cos(\theta) + A_z * \sin(\theta) \tag{1}$$
$$A_z{}' = A_z * \cos(\theta) - A_y * \sin(\theta) \tag{2}$$

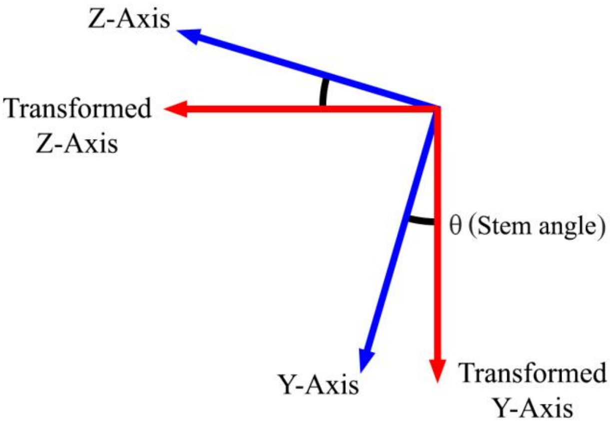


Figure 40 Axis transformation to compensate the stem angle

4.1.4 Road-facing video conversion

The PiCamera library records the videos in '.h264' format by default. Since the PiCamera has a history of working on mobile phones and which requires audio to be combined from different sources, the ability to record in '.mp4' format is missing [67]. The '.h264' format does not contain information about the frames per second recorded. Due to this all the widely available media players such as Video Lan Client (VLC), play the video either too fast or too slow and skip a few frames. This creates the need to convert the video to a '.mp4' format so the GUI or any other software can play the video seamlessly. FFMPEG is used for this conversion and can be done within the GUI. Upon clicking the Pre-Process button and thereby selecting a folder, all the '.h264' files in the folder will be converted to '.mp4' format and assigned relevant names.

4.1.5 Combining the uptime with UTC

As mentioned in section 2.2.2.4 the data is logged at an average of 10 Hz frequency. The UTC is collected with the second precision resulting in the loss of the exact timestamp of the data collection. Initially, this has been resolved by checking the data points per second and dividing them equally. This results in an unrealistic representation of the data. To compensate for this, the uptime of the e-scooter is combined with the UTC. The uptime being collected with millisecond precision when combined with the UTC gives an accurate representation of the time-series.

4.2 Subjective Data analysis

The main objective of carrying out the subjective data analysis has been to understand the effectiveness of the questionnaire. When the questionnaire is used as a part of the larger naturalistic data collection study, the responses should enable an understanding of the participant's demographics, experience with e-scooters, in addition to their opinions. Based on suggestions from experts who have a demonstrated history with naturalistic data collection, there have been three changes made. In the questions 'How often do you use following means of transportation?' and 'How often do you use e-scooters during the year?' a new option of 'less than 1 day per week' has been added in addition to the various other options as shown in the updated questionnaire in Appendix B. Question 10 i.e. 'Which mode of transport do you combine the journey with an e-scooter?' previously allowed only one option to be selected as a response has been changed to allow multiple options to be selected. The subjective data analysis although is mainly intended to be completed by the participants of the study, responses from non-participants have also been considered. Given the limited schedule resulting in limited participants, understanding the possible issues with regards to the questionnaire would not be effective with only the responses from participants.

4.3 Objective data analysis

As described in section 3, the major objective of the data collection has been to identify the potential issues with the logger and check the quality of the data collected. Thereby, the objective data analysis involves more understanding of each of the variables collected, the modifications possible to have an overall better setup for each of the specific variables.

4.3.1 Ride data

A total of 36 datasets have been recorded of which 4 are discarded. The reason for discarding has been explained further in section 4.3.4. A total of 131 km has been covered in the 32 datasets considered for analysis. The routes covered during all the rides combined are shown in Figure 41. As mentioned in section 3.2, the participants used the scooter to travel between home and work resulting in several journeys with overlapping route hence only 3 routes are marked on the graph with 3 different colors.

4.3.2 Trigger events

Out of the 32 datasets, 4 rides have trigger flags. The press of the trigger button is marked on the video as shown in Figure 42. Table 5 provides details on each of these trigger events. It is worth noting that in each of these 4 events the trigger flag is marked a few seconds after the occurrence of these uncomfortable situations for the rider. Hence the marking of the event in the graph is done from 5 seconds before and 5 seconds after the trigger flag as shown in Figure 35. The delay can be attributed to the complex nature of avoidance maneuvers which in hindsight may appear to be simple.

The complex set of tasks includes identifying the critical situation, identifying the parties involved, estimating the trajectories of the surrounding road users, planning the avoidance maneuver, carrying out the planned action all happening within a fraction of a second among several other possible tasks. The situation that led to the first trigger in the 2022_04_29_07_22 dataset is shown in Figure 43 and the trigger flag is shown in Figure 42 indicates the minor delay. Each of these flagged incidents has been reviewed along with the participants to confirm it. On Prototype-1 the flag is marked both on the video and the kinematic data while on Prototype-2 the flag is marked on the video only since the logging of the kinematic data is isolated from the Raspberry Pi.

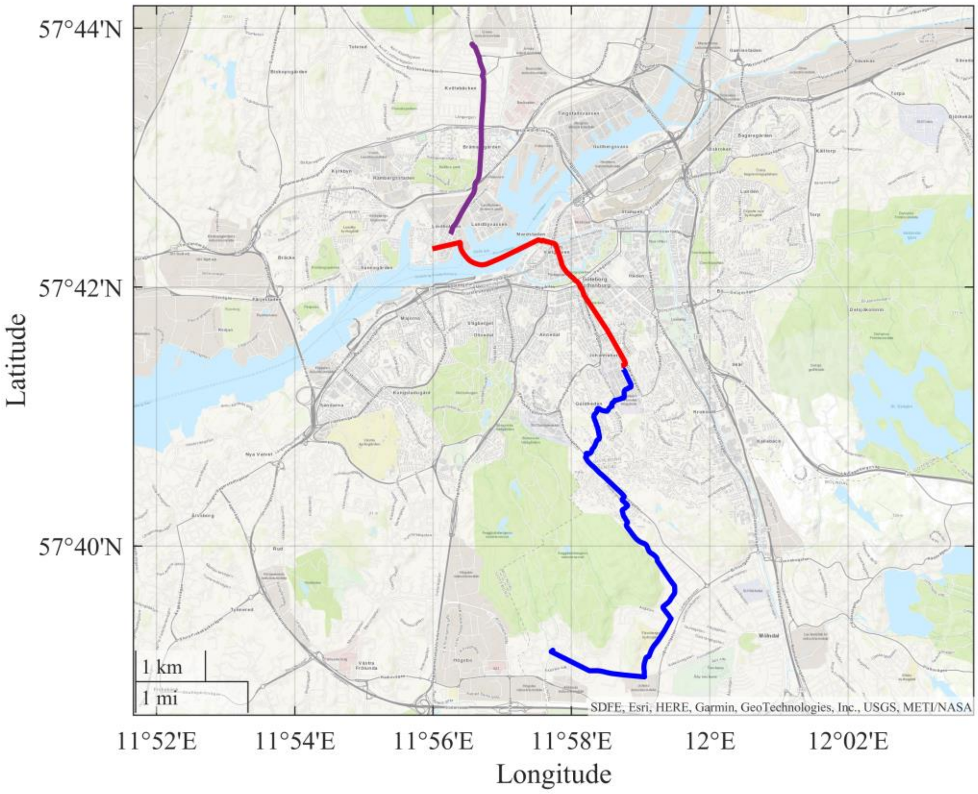


Figure 41 Map including the routes covered during the data collection phase

Table 5 Description of trigger events in the datasets

Dataset identifier	**Description of uncomfortable situations for the rider resulting in the trigger**
2022_04_29_07_22	A kick-scooter [68] cuts the path of the e-scooter rider unexpectedly resulting in sudden braking.
2022_04_29_07_22	A pedestrian steps in the bicycle path resulting in the e-scooter rider having to move to the oncoming traffic bicycle lane.
2022_04_26_13_44	A pedestrian is in the bicycle lane waiting to cross the street resulting in the e-scooter rider having to travel in the oncoming traffic bicycle lane
2022_05_16_10_29	The car travelling perpendicular to the path of the e-scooter brakes late resulting in the near-crash situation and the e-scooter rider using braking to avoid conflict.

Figure 42 Trigger indicator on the recorded video

Figure 43 Situation considered uncomfortable by rider in 2022_04_29_07_22 dataset

4.3.3 Ride feature identification from the data logged

Several ride features can be identified based on the data logged. Among them, two have been described in detail showcasing the potential of the collected data. With several other variables available, feature detection can be further expanded to detect bad riding behavior among others. These can also be used for modelling the rider and the e-scooter. Hansson and Congreve Lifh [69] in their thesis work have identified several anomaly detections in e-scooter rider behaviors based on similar data.

4.3.3.1 Cobblestone riding detection

Riding on the cobblestone at high speed can result in loss of balance causing single crashes. The vibrations resulting from riding on cobblestones are evident in the accelerometer readings on all three axes. Among these readings, the X-axis accelerometer readings provide a clear indication. This has been identified in multiple ride data and is confirmed by the video data collected using the road-facing camera. Accelerometer readings from one of the rides are shown in Figure 44 where the yellow box indicates the part of the ride where the rider is on the cobblestone surface. This drastic deviation can be linked to the rider constantly adjusting the handlebar to maintain the balance.

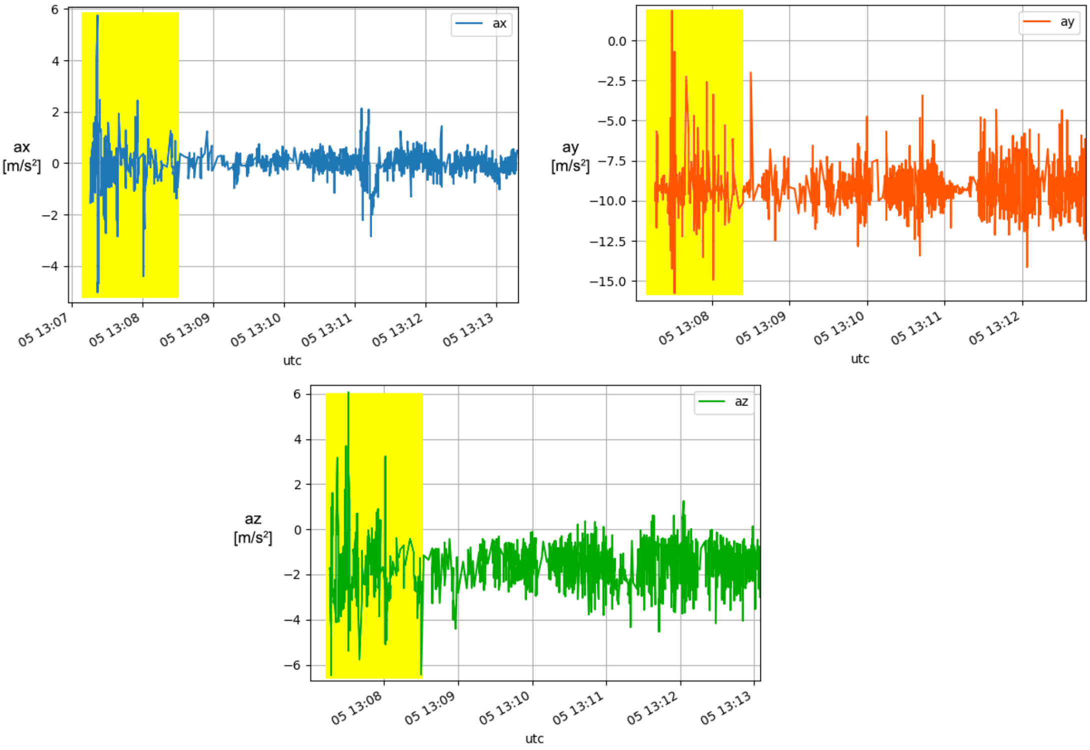


Figure 44 Accelerometer data in X-axis from a dataset.

4.3.3.2 Harsh braking

Harsh braking is when the rider applies braking force significantly higher than normal over a short period. This can be a surrogate measure of the occurrence of a safety-critical event. The brakes on each e-scooter are unique and vary based on the time from the last service and their wear. The variables ‘brake_l’ and ‘brake_r’ indicate the position of the lever, which on a recently serviced e-scooter can have a lower value but provide the same braking force as that of the e-scooter serviced a fortnight ago which reads a higher value. Thus, fixing an absolute value as a threshold is not an accurate solution. Based on average values of the ‘brake_l’ and ‘brake_r’ variables during a trip, a threshold is set for each of these and a range indicating the normal operation window.

Figure 45 shows the values of the ‘brake_l’ variable in a dataset. The average value during braking maneuver is calculated to be 72. To identify the range of safe operation the difference between the mean (70 in this case) and minimum value (48 in this dataset) is determined and the same (72 - 48 = 24) is considered on either side of the mean. Thus, the overall range of safe operation extends from 48 to 96. Some margin is provided to account for any aggressive riding behaviors which is marked with the grey color in the graph and any value above this is considered as harsh braking. While several methods can be employed to determine the region of safe operation, a simple method has been described here due to the limited time available for investigation of more sophisticated methodology. Further analyses can be performed when more data is available.

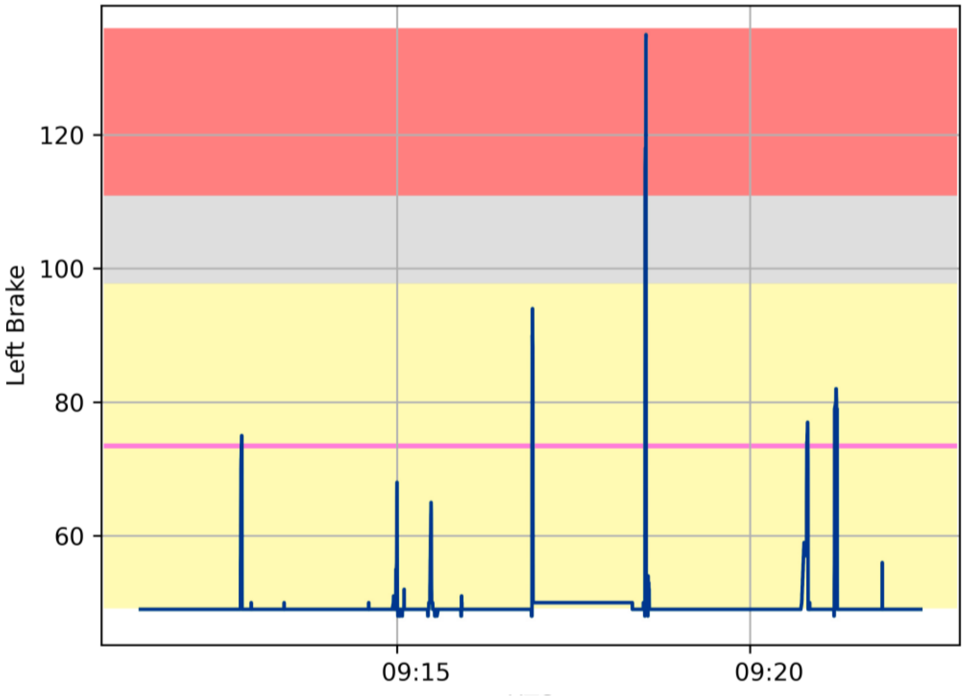


Figure 45 Plot of the position of the left brake lever in a dataset

4.3.4 Missing data points

As mentioned previously, 4 of the datasets have been discarded. The IoT of the e-scooter can store data from only one ride and uploads it to the cloud storage, with upload taking an average of less than a minute after the end of the ride. If the user/participant unlocks the e-scooter during this timeframe the upload process is terminated, and the file containing the data is automatically deleted. This issue has resulted in incomplete datasets being uploaded which are discarded from further analysis.

5 Discussions and Conclusions

Prototype-0 developed by Schmidt et al. [47] proved that the e-scooters can be equipped with a data logger for the collection of ND. However, the setup proved to be bulky and expensive. Prototype-2 developed as a part of this thesis has showcased a data logger that can be scaled to a large fleet and is robust enough to handle the conditions the e-scooter is exposed to on a daily basis. The Python-based GUI has proven to be more accessible and feature-rich compared to the MATLAB based predecessor with the videos played simultaneously and several features such as the turn indicator status. The pilot dataset collected and the preliminary analysis carried out as a part of the thesis have demonstrated the potential of the data collected by using the setup on a large fleet. The Raspberry Pi based setup is expected to be implanted onto a small fleet of e-scooters as a part of the E-safe pre-study [70] taking place during the summer of 2022.

5.1 Prototype e-scooter

While Prototype-0 has proven to be effective in data logging, there have been several issues including the external power supply and an overall bulky setup. The Prototype-2 developed as in this dissertation has not only solved the issues but also developed the setup to be more robust and harmonious with the existing peripherals on the e-scooter. The addition of a voltage converter has enabled powering the setup from the battery of the e-scooter which provides a longer duration of data logging as compared to its predecessors. The establishment of a UART protocol to receive messages from IoT has enabled the elimination of J-Link which not only reduces the cost but also removes the uncertainties involved with the J-Link debugger tool in the communication thread. The prototypes in this thesis have undergone extensive testing. Apart from the 131 km covered during the data collection stage, these test e-scooters have covered a distance of 226.5 km during various stages of testing resulting in a total of 357.5 km covered. The tests ranged from checking the structural integrity of the casing to the bugs in the software involved in logging and data upload. While several issues have been encountered in the testing stages of the prototype e-scooter, each of these has been addressed. With the logger having performed with minimal issues during the data collection stage, it has proven to be a reliable setup. Thus, it can be expanded to a bigger fleet with high confidence for conducting NRS.

The trigger flag has been an essential tool in the identification of events of interest from the perspective of traffic safety. While the kinematic data would provide some insight into how the e-scooterist handled the situation it fails to give an understanding of the reactions of the surrounding road user. It also fails to completely represent the aspects that caused the unsafe events. The road-facing camera with the wide FoV has proven to be very effective in this aspect. The wide FoV although results in fisheye distortion, is essential in situations such as an intersection, where the vehicles are travelling perpendicular to each other. While the 220° FoV provides significant information about the surroundings, it creates an optical illusion where an object appears farther away than it actually is. Complementing the road-facing camera, rider-facing helps identify the reactions of the rider in each of these scenarios. Neither the road-facing camera nor the kinematics alone provides sufficient data to understand the reasons for delayed reaction or any specific maneuvers made by the rider making rider-facing cameras essential. It is also fundamental to understand the condition of the rider before the occurrence of the unsafe event (e.g., identification of a distracted rider). Having a single timestamp on the dataset and videos has enabled the syncing of the video with the data as described in section 4.1.2. As a consequence of having kinematic data-logging done on the IoT and no communication from the Raspberry Pi module to the IoT exists, the trigger flag is not marked on the kinematic data. Having this flag on the kinematic data will make the whole process of data analysis easy and effective.

The prototype developed in the thesis is an independent unit and can be mounted on any of the Voi e-scooters, with modification to the firmware on the IoT required for communication with the logger. This adds flexibility to the setup and enables the data collection from different cities to be carried out with little effort.

The Raspberry Pi based data logger module is not only limited to ND collection. Due to the increasing safety concerns, several cities have restricted the shared micro-mobility service operators with several stringent requirements put forth [71], [72]. Chicago for instance has made sidewalk riding detection a criterion for selecting the possible operators in the city [73]. The data logger developed as a part of the thesis, with some additional computer vision libraries, could be used to detect various aspects such as jumping traffic lights, sidewalk riding etc. in addition to collecting naturalistic riding data. If a communication protocol is established with the IoT of the e-scooter, this module can thereby communicate with the IoT and take appropriate action such as slowing down the e-scooter.

5.2 Data Analysis

Although the major objective of the thesis has been to develop a ride data logger, a pilot data collection and analysis have been conducted to showcase the potential of the dataset when collected on a large scale and debug potential issues.

Several studies have illustrated the development of a driver model and the potential for counterfactual simulation based on the naturalistic data collected in the studies such as the SHRP2 [74]–[78]. Section 4.3.3 showcased two of the possible ride-feature detection. With several more possible feature identifications, a rider model can be developed by combining each of these variables. With the rider models, several safety systems can be implemented on both the e-scooter and surrounding road users. By estimating the trajectory of an e-scooterist, warning systems can be employed on cars and trucks. This system can assist drivers to avoid that safety-critical situations with e-scooter riders devolve into crashes. These features on an individual level can also help regulate the e-scooters and the riders. For instance, reduction of the speed based on riding behaviors or deploying the regenerative braking to provide additional braking force when harsh braking conditions are detected. The details of the kinematics can be used for reconstruction of the scenario and thereby develop counterfactual simulations of various measures, for instance the deployment of an ABS and its effect on that given scenario.

The dataset also helps in the identification of the crash causation mechanism. As described in section 4.3.2 when each event in a large set of data is annotated, the events that develop to cause the crash can be identified. This has been proven to be an effective method in crash mechanism estimation using ND [79]–[81]. Uchida et al. [82] in their study have investigated the factors contributing to crashes on cars in Japan using naturalistic data. Similar studies to understand the rear-end crashes on commercial vehicles in China have been carried out by Piccinini et al. [79]. Based on the outcome of conducting these analyses on the dataset collected in a larger NRS, the riders of e-scooters can be better trained to avoid the situations.

In addition to the aforementioned analysis, several statistical evaluations can be carried out as a part of the data analysis. There have been numerous methods used in the statistical analysis of data from the 100-car naturalistic driving study and SHRP2 study. This spans from the study by Dingus et al. [83] to identify the risky factors in driving to the study by Guo and Fang [84] to assess the individual driver risk based on naturalistic driving data. In addition to these, odds-ratio has been a common tool opted to understand the likeliness of a parameter to result in a crash or near-crash [85]–[87]. This methodology has been implemented in studies to understand the association between cell phones and crashes. Similarly, on an e-scooter, this can enable understanding the likeliness of the weather conditions or multiple people riding on an e-scooter resulting in a crash. Nevertheless with crashes being rare even in some of the largest naturalistic study datasets, obtaining statistically significant data to conduct this analysis may be hard.

5.3 Limitations and scope for future work

The implementation of the UART communication has made the overall setup robust but given it is one-way communication, limits the possibilities which would be available otherwise. Implementing a full-fledged SPI or I2C communication will enable two-way communication which thereby can help trigger flag marking on the kinematic data. Having two-way communication can also help remotely detect the status of the video recording by using the Voi IoT tools.

Given the video data is uploaded only when connected to a Wi-Fi, and the module does not include a sim card, each e-scooter must be transported to the office to attain the videos. This costs both time and money limiting the scalability. Having a sim card connected to the Pi Zero-2 and having internet access will not only allow data upload to happen remotely but also have fleet management software implemented on the module.

As described in section 4.3.4 there has been missing datasets due to the e-scooter being unlocked before the successful upload of the kinematic dataset. While this has not resulted in a significant loss in data, having the IoT store data for two rides will prevent data loss. The detailed analysis of the data collected is not carried out due to the limitation in the time available for the thesis work.

Due to time limitations and other constraints as described in section 3.2 only four participants within the university have been considered for data collection. The bias induced by this has been neglected since the bias does not affect the major goal of the development of the data logging setup. The testing after the sunset could not be carried out due to the long days during the summer.

The GUI can be developed further to allow the user to export a particular part of the ride along with several other features such as the possibility to zoom in the GPS plot.

6 References

[1] C. S. Smith and J. P. Schwieterman, “E-Scooter Scenarios: Evaluating the Potential Mobility Benefits of Shared Dockless Scooters in Chicago,” Chicago, 2018. [Online]. Available: https://las.depaul.edu/centers-and-institutes/chaddick-institute-for-metropolitan-development/research-and-publications/Documents/E-ScooterScenariosMicroMobilityStudy_FINAL_20181212.pdf.

[2] S. Shaheen, A. Cohen, and I. Zohdy, “Shared Mobility: Current Practices and Guiding Principles,” Washington, DC, 2016. [Online]. Available: https://ops.fhwa.dot.gov/publications/fhwahop16022/fhwahop16022.pdf.

[3] A. J. Hawkins, “The electric scooter craze is officially one year old — what’s next? - The Verge,” *The Verge*, Sep. 20, 2018. https://www.theverge.com/2018/9/20/17878676/electric-scooter-bird-lime-uber-lyft (accessed Jan. 30, 2022).

[4] M. C. Lesh, “Innovative Concepts in First-Last Mile Connections to Public Transportation,” in *Proceedings of the 3rd International Conference on Urban Public Transportation Systems*, 2013, pp. 63–74, doi: 10.1061/9780784413210.007.

[5] H. Wang and A. Odoni, “Approximating the Performance of a ‘Last Mile’ Transportation System,” *Transp. Sci.*, vol. 50, no. 2, pp. 659–675, Sep. 2016, doi: 10.1287/TRSC.2014.0553.

[6] B. Berg Wincent, “Gångavstånd för resor med elsparkcykel: En studie baserat på elsparkcykelaktören Voi Technology i Stockholm,” KTH, 2019.

[7] “Max Plus X - Segway Commercial.” https://b2b.segway.com/max-plus-x/ (accessed Feb. 03, 2022).

[8] S. Gössling, “Integrating e-scooters in urban transportation: Problems, policies, and the prospect of system change,” *Transp. Res. Part D Transp. Environ.*, vol. 79, p. 102230, Feb. 2020, doi: 10.1016/J.TRD.2020.102230.

[9] “Shared Micromobility in the U.S.: 2018 | National Association of City Transportation Officials.” https://nacto.org/shared-micromobility-2018/ (accessed Feb. 03, 2022).

[10] “Shared Micromobility in the U.S.: 2019 | National Association of City Transportation Officials.” https://nacto.org/shared-micromobility-2019/ (accessed Feb. 03, 2022).

[11] A.-K. Ahlmer, C. Fredricsson, E. Adell, and C. Ljungberg, “Elsparkcyklar i delningsekonomin,” Stockholm, 2019. [Online]. Available: https://www.regionstockholm.se/globalassets/4.-regional-utveckling/publicerade-dokument/rapport-elsparkcyklar_200305.pdf.

[12] “E-Scooter-sharing - Sweden | Statista Market Forecast.” https://www.statista.com/outlook/mmo/mobility-services/e-scooter-sharing/sweden#global-comparison (accessed Feb. 03, 2022).

[13] “Voi Scooter Cities.” https://www.voiscooters.com/locations/ (accessed Jan. 30, 2022).

[14] P. Mattsson, “Stockholms besked till elscooter-bolagen: De som inte sköter sig åker ut,” *Dagens industri*, Dec. 16, 2021. https://www.di.se/digital/stockholms-besked-till-elscooter-bolagen-de-som-inte-skoter-sig-aker-ut/ (accessed Feb. 03, 2022).

[15] S. Gorjifar, “Sammanställning av resultat av Sommarjobb Uppdrag 2021,” Göteborg, Sweden, 2021. [Online]. Available: https://goteborg.se/wps/wcm/connect/8e5193ae-0567-46dd-af4d-

853f53c018ac/20211108+Företagssammanställning+sommarjobbsuppdrag.pdf?MOD=AJPERES.

[16] "Voi's year in brief: through headwinds, tailwinds and massive growth." https://www.voiscooters.com/blog/end-of-year-2021/ (accessed Feb. 03, 2022).

[17] Transportstyrelsen, "Utredning behov av förenklade regler för eldrivna enpersonsfordon," 2021. [Online]. Available: https://www.transportstyrelsen.se/globalassets/global/publikationer/vag/slutrapport-utredning-regler-eldrivna-enpersonsfordon.pdf.

[18] D. García-Vallejo, W. Schiehlen, and A. García-Agúndez, "Dynamics, Control and Stability of Motion of Electric Scooters," in *Advances in Dynamics of Vehicles on Roads and Tracks*, 2020, pp. 1199–1209, doi: 10.1007/978-3-030-38077-9_139.

[19] J. Todd, D. Krauss, J. Zimmermann, and A. Dunning, "Behavior of Electric Scooter Operators in Naturalistic Environments," *SAE Tech. Pap.*, vol. 01, no. 1007, Apr. 2019, doi: 10.4271/2019-01-1007.

[20] N. K. Namiri, H. Lui, T. Tangney, I. E. Allen, A. J. Cohen, and B. N. Breyer, "Electric Scooter Injuries and Hospital Admissions in the United States, 2014-2018," *JAMA Surg.*, vol. 155, no. 4, pp. 357–359, Apr. 2020, doi: 10.1001/JAMASURG.2019.5423.

[21] A. Santacreu, "Safe Micromobility," 2020. [Online]. Available: https://itf-oecd.org/sites/default/files/docs/safe-micromobility_1.pdf.

[22] A. B. Brownson, P. V. Fagan, S. Dickson, and I. D. Civil, "Electric scooter injuries at Auckland City Hospital.," *N. Z. Med. J.*, vol. 132, no. 1505, pp. 62–72, Nov. 2019, Accessed: Jan. 30, 2022. [Online]. Available: https://assets-global.website-files.com/5e332a62c703f653182faf47/5e332a62c703f648f12fccca_BrownsonFINAL.pdf.

[23] H. Stigson and M. Klingegård, "Kartläggning av olyckor med elsparkcyklar och hur olyckorna kan förhindras," 2019. [Online]. Available: https://www.trafikverket.se/contentassets/c65cbf975681419d87ec9bba1fa4470d/folksam-rapport-elsparkcyklar.pdf.

[24] H. Stigson, I. Malakuti, and M. Klingegård, "Electric scooters accidents: Analyses of two Swedish accident data sets," *Accid. Anal. Prev.*, vol. 163, p. 106466, Dec. 2021, doi: 10.1016/J.AAP.2021.106466.

[25] T. K. Trivedi *et al.*, "Injuries Associated With Standing Electric Scooter Use," *JAMA Netw. Open*, vol. 2, no. 1, Jan. 2019, doi: 10.1001/JAMANETWORKOPEN.2018.7381.

[26] S. N. F. Blomberg, O. C. M. Rosenkrantz, F. Lippert, and H. Collatz Christensen, "Injury from electric scooters in Copenhagen: a retrospective cohort study," *BMJ Open*, vol. 9, no. 12, Dec. 2019, doi: 10.1136/BMJOPEN-2019-033988.

[27] L. M. Kobayashi *et al.*, "The e-merging e-pidemic of e-scooters," *Trauma Surg. Acute Care Open*, vol. 4, no. 1, Jul. 2019, doi: 10.1136/TSACO-2019-000337.

[28] B. Trivedi, M. J. Kesterke, R. Bhattacharjee, W. Weber, K. Mynar, and L. V. Reddy, "Craniofacial Injuries Seen With the Introduction of Bicycle-Share Electric Scooters in an Urban Setting," *J. Oral Maxillofac. Surg.*, vol. 77, no. 11, pp. 2292–2297, Nov. 2019, doi: 10.1016/j.joms.2019.07.014.

[29] M. Aizpuru, K. X. Farley, J. C. Rojas, R. S. Crawford, T. J. Moore, and E. R. Wagner, "Motorized scooter injuries in the era of scooter-shares: A review of the national electronic surveillance system," *Am. J. Emerg. Med.*, vol. 37, no. 6,

pp. 1133–1138, Jun. 2019, doi: 10.1016/J.AJEM.2019.03.049.

[30] F. Guo, S. G. Klauer, M. T. McGill, and T. A. Dingus, "Evaluating the Relationship Between Near-Crashes and Crashes: Can NearCrashes Serve as a Surrogate Safety Metric for Crashes?," Blacksburg, Virginia, 2010. [Online]. Available: https://www.nhtsa.gov/sites/nhtsa.gov/files/documents/dot_hs_811_382.pdf.

[31] T. A. Dingus *et al.*, "The 100-Car Naturalistic Driving Study, Phase II - Results of the 100-Car Field Experiment," Blacksburg, Virginia, 2006. [Online]. Available: http://www.nhtsa.gov/DOT/NHTSA/NRD/Multimedia/PDFs/Crash Avoidance/Driver Distraction/100CarMain.pdf.

[32] "SHRP 2 Naturalistic Driving Study." https://www.shrp2nds.us/ (accessed Jan. 31, 2022).

[33] "euroFOT - European Field Operational Test on Active Safety Systems." https://www.eurofot-ip.eu/ (accessed Jan. 31, 2022).

[34] "UDRIVE - European Naturalistic Driving Study ." https://wiki.fot-net.eu/index.php/UDRIVE (accessed Jan. 31, 2022).

[35] K. Custer, "100-Car Data," vol. 2. Virginia Tech Transportation Institute (VTTI), Jul. 06, 2018, doi: 10.15787/VTT1/CEU6RB.

[36] M. Dozza and A. Fernandez, "Understanding bicycle dynamics and cyclist behavior from naturalistic field data (November 2012)," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 376–384, Feb. 2014, doi: 10.1109/TITS.2013.2279687.

[37] M. Dozza, G. F. Bianchi Piccinini, and J. Werneke, "Using naturalistic data to assess e-cyclist behavior," *Transp. Res. Part F Traffic Psychol. Behav.*, vol. 41, pp. 217–226, Aug. 2016, doi: 10.1016/J.TRF.2015.04.003.

[38] L. Gustafsson and J. Archer, "A naturalistic study of commuter cyclists in the greater Stockholm area," *Accid. Anal. Prev.*, vol. 58, pp. 286–298, Sep. 2013, doi: 10.1016/J.AAP.2012.06.004.

[39] M. Johnson, J. Charlton, J. Oxley, and S. Newstead, "Naturalistic Cycling Study: Identifying Risk Factors for On-Road Commuter Cyclists," *Ann. Adv. Automot. Med. / Annu. Sci. Conf.*, vol. 54, p. 275, 2010, Accessed: Feb. 02, 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3242541/.

[40] F. W. Siebert, M. Ringhand, F. Englert, M. Hoffknecht, T. Edwards, and M. Rötting, "Braking bad – Ergonomic design and implications for the safe use of shared E-scooters," *Saf. Sci.*, vol. 140, p. 105294, Aug. 2021, doi: 10.1016/J.SSCI.2021.105294.

[41] R. Sherony, "e-Scooters: A Roadmap to Vehicle Safety - Toyota USA Newsroom," *Toyota pressroom*, Jun. 03, 2020. https://pressroom.toyota.com/e-scooters-a-roadmap-to-vehicle-safety/ (accessed Feb. 13, 2022).

[42] S. S. Betrabet, "Data Acquisition and Processing Pipeline for E-Scooter Tracking Using 3d Lidar and Multi-Camera Setup," Purdue University, 2020.

[43] K. Apurv, R. Tian, and R. Sherony, "Detection of E-scooter Riders in Naturalistic Scenes," *Comput. Vis. Pattern Recognit.*, Nov. 2021, Accessed: Feb. 13, 2022. [Online]. Available: https://arxiv.org/abs/2111.14060v1.

[44] A. Violin, "Development of an experimental protocol for testing new electric personal mobility vehicles," Chalmers University of Technology, 2020.

[45] C. Garman, S. G. Como, I. C. Campbell, J. Wishart, K. O'Brien, and S. McLean, "Micro-Mobility Vehicle Dynamics and Rider Kinematics during Electric Scooter Riding," *SAE Tech. Pap. 2020-01-0935*, vol. 1, Apr. , doi: 10.4271/2020-01-0935.

[46] P. Brunner, A. Locken, F. Denk, R. Kates, and W. Huber, “Analysis of experimental data on dynamics and behavior of e-scooter riders and applications to the impact of automated driving functions on urban road safety,” in *IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 219–225, doi: 10.1109/IV47402.2020.9304835.

[47] D. Schmidt, F. Lindelöw, G. Gunashekara, and R. R. Pai, “Pilot Naturalistic Riding Study (NRS) with VOI e-scooters to improve traffic safety,” Göteborg, Sweden, 2022. [Online]. Available: https://hdl.handle.net/20.500.12380/304489.

[48] “Buy a Raspberry Pi 3 Model B – Raspberry Pi.” https://www.raspberrypi.com/products/raspberry-pi-3-model-b/ (accessed Feb. 25, 2022).

[49] “Raspberry Pi Zero 2 W – Raspberry Pi.” https://www.raspberrypi.com/products/raspberry-pi-zero-2-w/ (accessed Feb. 25, 2022).

[50] “Köp Raspberry Pi Zero 2 W till rätt pris @ Electrokit.” https://www.electrokit.com/produkt/raspberry-pi-zero-2-w/?src=raspberrypi (accessed Feb. 25, 2022).

[51] “Köp Raspberry Pi 3 1GB Model B till rätt pris @ Electrokit.” https://www.electrokit.com/produkt/raspberry-pi-3-1gb-model-b/ (accessed Feb. 25, 2022).

[52] PTC Creo, “Creo Parametric 5.0.0.0.” PTC, Boston, USA.

[53] “Meet Voiager 5, Voi’s safest and most durable e-scooter to date.” https://www.voiscooters.com/blog/voiager-5/ (accessed May 07, 2022).

[54] “Pololu 3.3V Step-Up Voltage Regulator U1V10F3.” https://www.pololu.com/product/2563 (accessed Feb. 25, 2022).

[55] “TEL 10-4811 | Traco Power.” https://www.tracopower.com/model/tel-10-4811 (accessed Feb. 25, 2022).

[56] “Odseven Fisheye ZeroCam for Raspberry Pi Zero.” https://odseven.com/products/odseven-camera-for-raspberry-pi-zero-with-big-camera-fisheye-170-degree (accessed Apr. 11, 2022).

[57] “LS-32220 Arducam 220 FoV Lens.” https://www.arducam.com/downloads/Lenses/Fisheye/LS-32220.pdf (accessed Apr. 11, 2022).

[58] “Arducam 5MP OV5647 NoIR Camera Board /w M12x0.5 mount for Raspberry Pi 4/3B+/3 Camera - Arducam.” https://www.arducam.com/product/arducam-ov5647-noir-m12x0-5-mount-camera-board-w-raspberry-pi/ (accessed May 07, 2022).

[59] “Arducam Fisheye Low Light USB Camera with Mini Metal Case, 2MP 1080P IMX291 Wide Angle Mini H.264 UVC Video Camera Board with Microphone for Computer, Laptop, and Desktop - Arducam.” https://www.arducam.com/product/arducam-imx291-usb-camera-with-case-b026101/ (accessed May 04, 2022).

[60] “Multithreading (computer architecture) - Wikipedia.” https://en.wikipedia.org/wiki/Multithreading_(computer_architecture) (accessed Apr. 09, 2022).

[61] “Cryptsetup and LUKS,” 2020. https://gitlab.com/cryptsetup/cryptsetup (accessed Jun. 19, 2022).

[62] “Encryption - General Data Protection Regulation (GDPR).” https://gdpr-info.eu/issues/encryption/ (accessed Jun. 19, 2022).

[63] N. Craig-Wood, “Rclone.” Surrey, UK, [Online]. Available: https://rclone.org/.

[64] P. Vixie, “Crontab(5).” Internet Systems Consortium, 2004, [Online]. Available: https://github.com/vixie/cron.

[65] H. P. Anvin, “flock(1).” 2006, [Online]. Available: https://git.zytor.com/historic/flock.git/.

[66] OpenStreetMap contributors, “OpenStreetMap.” 2022, [Online]. Available: https://www.openstreetmap.org.

[67] “5. Frequently Asked Questions (FAQ) — Picamera 1.13 Documentation.” https://picamera.readthedocs.io/en/release-1.13/faq.html#why-is-playback-too-fast-too-slow-in-vlc-omxplayer-etc (accessed May 13, 2022).

[68] “Kick sccoter definition.” https://en.wikipedia.org/wiki/Kick_scooter (accessed May 19, 2022).

[69] I. Hansson and J. Congreve Lifh, “Anomaly Detection in Riding Behaviours : Using Unsupervised Machine Learning Methods on Time Series Data from Micromobility Services,” Umeå University, 2022.

[70] “E-safe pre-study | Vinnova.” https://www.vinnova.se/en/p/e-safe-pre-study/ (accessed May 25, 2022).

[71] J. Bonnet, “Paris: Voici les trois opérateurs de trottinettes retenus par la Mairie,” *BFM Business*, Jul. 23, 2020.

[72] C. Duxbury, “Stockholm’s clampdown on ‘chaotic’ e-scooters – POLITICO,” *Politico*, Stockholm, Jan. 17, 2022.

[73] L. Lightfoot, K. Meyer, and G. Biagi, “Scooter Sharing Business - Draft rules,” Chicago, 2022. [Online]. Available: https://www.chicago.gov/content/dam/city/depts/cdot/Misc/EScooters/2022/Chicago Scooter License Rules and Regulations_01.12.22 - DRAFT rev.pdf.

[74] M. M. Ahmed and A. Ghasemzadeh, “The impacts of heavy rain on speed and headway Behaviors: An investigation using the SHRP2 naturalistic driving study data,” *Transp. Res. Part C Emerg. Technol.*, vol. 91, pp. 371–384, Jun. 2018, doi: 10.1016/J.TRC.2018.04.012.

[75] A. Das, A. Ghasemzadeh, and M. M. Ahmed, “Analyzing the effect of fog weather conditions on driver lane-keeping performance using the SHRP2 naturalistic driving study data,” *J. Safety Res.*, vol. 68, pp. 71–80, Feb. 2019, doi: 10.1016/J.JSR.2018.12.015.

[76] A. Ghasemzadeh and M. M. Ahmed, “Quantifying regional heterogeneity effect on drivers’ speeding behavior using SHRP2 naturalistic driving data: A multilevel modeling approach,” *Transp. Res. Part C Emerg. Technol.*, vol. 106, pp. 29–40, Sep. 2019, doi: 10.1016/J.TRC.2019.06.017.

[77] J. Bärgman, V. Lisovskaja, T. Victor, C. Flannagan, and M. Dozza, “How does glance behavior influence crash and injury risk? A ‘what-if’ counterfactual simulation using crashes and near-crashes from SHRP2,” *Transp. Res. Part F Traffic Psychol. Behav.*, vol. 35, pp. 152–169, Nov. 2015, doi: 10.1016/J.TRF.2015.10.011.

[78] J. Bärgman, C. N. Boda, and M. Dozza, “Counterfactual simulations applied to SHRP2 crashes: The effect of driver behavior models on safety benefit estimations of intelligent safety systems,” *Accid. Anal. Prev.*, vol. 102, pp. 165–180, May 2017, doi: 10.1016/J.AAP.2017.03.003.

[79] G. Bianchi Piccinini, J. Engström, J. Bärgman, and X. Wang, “Factors contributing to commercial vehicle rear-end conflicts in China: A study using on-board event data recorders,” *J. Safety Res.*, vol. 62, pp. 143–153, Sep. 2017, doi: 10.1016/J.JSR.2017.06.004.

[80] L. Pipkorn and G. Bianchi Piccinini, “The role of off-path glances: A quantitative analysis of rear-end conflicts involving Chinese professional truck

drivers as the striking partners," *J. Safety Res.*, vol. 72, pp. 259–266, Feb. 2020, doi: 10.1016/J.JSR.2019.12.023.

[81] H. Hekla Eiríksdóttir, "Quantitative analysis of rear-end crash causation mechanisms based on naturalistic crash data," Chalmers University of Technology, Gothenburg, Sweden, 2016.

[82] N. Uchida, M. Kawakoshi, T. Tagawa, and T. Mochida, "An investigation of factors contributing to major crash types in Japan based on naturalistic driving data," *IATSS Res.*, vol. 34, no. 1, pp. 22–30, Jul. 2010, doi: 10.1016/J.IATSSR.2010.07.002.

[83] T. A. Dingus *et al.*, "Driver crash risk factors and prevalence evaluation using naturalistic driving data," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 113, no. 10, pp. 2636–2641, Mar. 2016, doi: 10.1073/PNAS.1513271113/.

[84] F. Guo and Y. Fang, "Individual driver risk assessment using naturalistic driving data," *Accid. Anal. Prev.*, vol. 61, pp. 3–9, Dec. 2013, doi: 10.1016/J.AAP.2012.06.014.

[85] S. G. Klauer, F. Guo, B. G. Simons-Morton, M. C. Ouimet, S. E. Lee, and T. A. Dingus, "Distracted Driving and Risk of Road Crashes among Novice and Experienced Drivers," *N. Engl. J. Med.*, vol. 370, no. 1, pp. 54–59, Jan. 2014, doi: 10.1056/NEJMSA1204142.

[86] A. Bálint, C. A. C. Flannagan, A. Leslie, S. Klauer, F. Guo, and M. Dozza, "Multitasking additional-to-driving: Prevalence, structure, and associated risk in SHRP2 naturalistic driving data," *Accid. Anal. Prev.*, vol. 137, p. 105455, Mar. 2020, doi: 10.1016/J.AAP.2020.105455.

[87] J. M. Owens, T. A. Dingus, F. Guo, Y. Fang, M. Perez, and J. Mcclafferty, "Crash Risk of Cell Phone Use While Driving: A Case-Crossover Analysis of Naturalistic Driving Data," Blacksburg, Virginia, Jan. 2018. Accessed: May 25, 2022. [Online]. Available: https://aaafoundation.org/wp-content/uploads/2018/01/CellPhoneCrashRisk_FINAL.pdf.

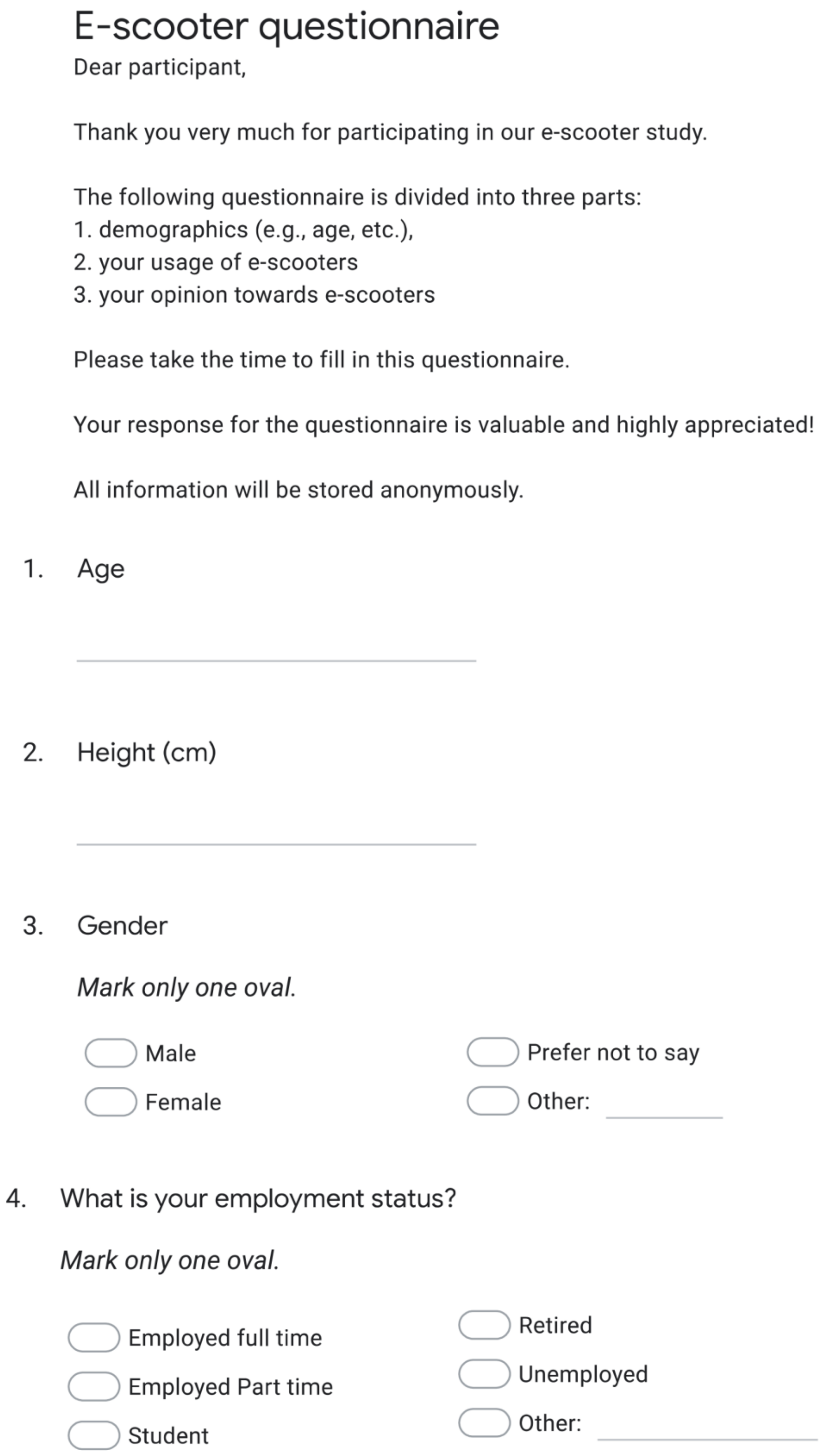
Appendix

A. Data upstream management

```python
# ================================================================= #
# Code to upload data using Rclone
# @ Author: Rahul Rajendra Pai <prahul@student.chalmers.se>
#================================================================== #

import subprocess
import time
import RPi.GPIO as GPIO


# Function to upload the file using rclone
def run_command(command):
    try:
        p = subprocess.check_output(command,
           shell=True).decode().rstrip()
    except subprocess.CalledProcessError as err:
        # Dry run returns error code 9:
        # Detailed error code https://rclone.org/docs/#exit-code
        if len(command.split('dry-run')) > 1:
            run_success = 9
            print('Error code: ', err.returncode)
        else:
            run_success = err.returncode
    # For the upload run we enter the else and return success
    else:
        print('We have run this once')
        if len(command.split('dry-run')) > 1:
            run_success = 9
        else:
            run_success = 0
    return run_success

# To check if the pi is connected to the Wifi or not before upload.
def getSSID():
    try:
        wifi_name = subprocess.check_output(
            ["/sbin/iwgetid -r"], shell=True).decode().rstrip()
    except subprocess.CalledProcessError:
        wifi_name = ''
    return wifi_name

# Setting up the GPIO Pins of the RGB LED
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(36, GPIO.OUT)  # Red LED
GPIO.setup(40, GPIO.OUT)  # Blue LED
```

```
 # Upload status to avoid rclone trying to upload even after a
successful upload
 upload_status = 0

 while True:
    # When the Pi is connected to the wifi: Uploading process
    if upload_status == 0:

       # Check if the Pi is connected to the internet
       if getSSID() == '':
          GPIO.output(36, GPIO.LOW)
          GPIO.output(40, GPIO.LOW)
          quit()

       # Turn the red LED on to indicate the start of the upload
process
       GPIO.output(36, GPIO.HIGH)

       # Dry Run
       dry_run_command = 'rclone copy -P --dry-run --error-on-no-
transfer /home/pi/voi_cam_data/ voi_cam:NRS/NRS_Log_data/8L3G'
       dry_run_output = run_command(dry_run_command)
       time.sleep(5)

       #Dry run returns 9 as the result
       if dry_run_output == 9:
          # Upload files
          upload_command = 'rclone move -P --error-on-no-
transfer --low-level-retries 1 /home/pi/voi_cam_data/
voi_cam:NRS/NRS_Log_data/8L3G'
          upload_run = run_command(upload_command)
          time.sleep(5)

          # Upload run can return 0 or 9 based on availability
of the files.
          if upload_run == 0 or upload_run == 9:
             GPIO.output(36, GPIO.LOW)
             GPIO.output(40, GPIO.HIGH)
             upload_status = 1
             print("Upload status:", upload_status)

    # Pi is connected to the wifi and the upload is complete
    elif upload_status == 1:
       if getSSID() == '':
          GPIO.output(36, GPIO.LOW)
          GPIO.output(40, GPIO.LOW)
          quit()
```

B. Subjective data collection questionnaire



E-scooter questionnaire

Dear participant,

Thank you very much for participating in our e-scooter study.

The following questionnaire is divided into three parts:
1. demographics (e.g., age, etc.),
2. your usage of e-scooters
3. your opinion towards e-scooters

Please take the time to fill in this questionnaire.

Your response for the questionnaire is valuable and highly appreciated!

All information will be stored anonymously.

1. Age

2. Height (cm)

3. Gender

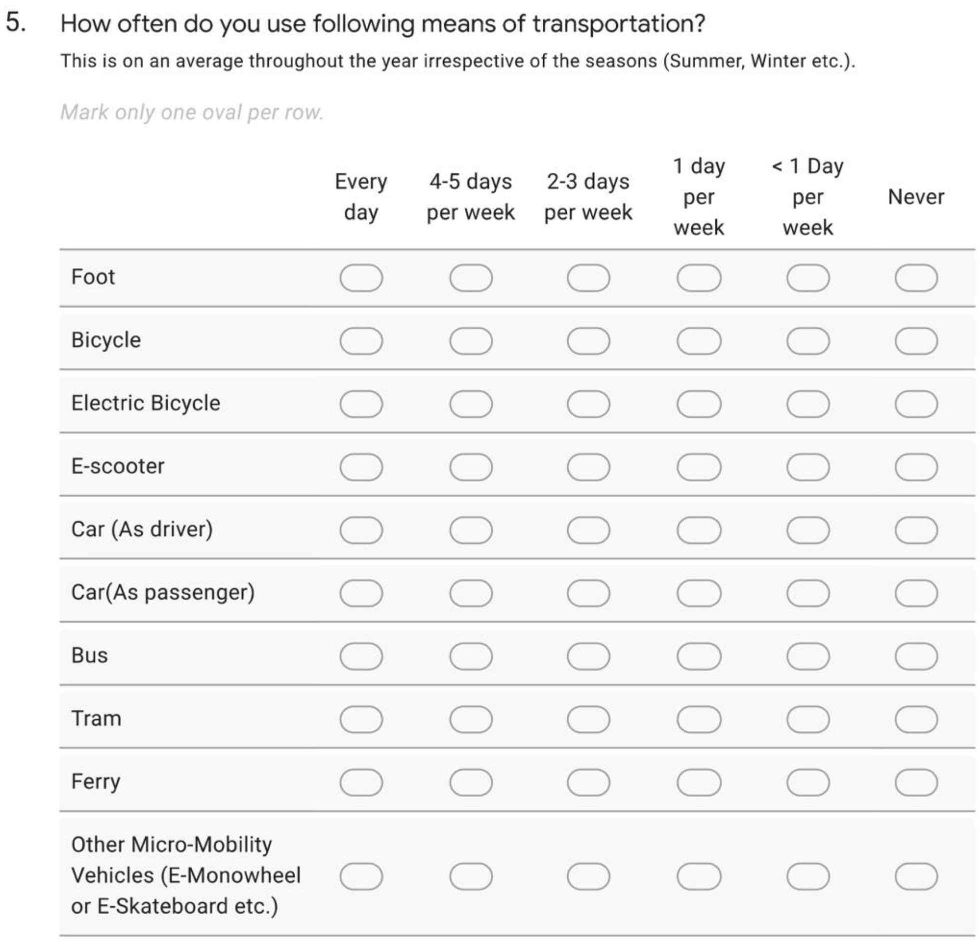
Mark only one oval.

○ Male
○ Female
○ Prefer not to say
○ Other:

4. What is your employment status?

Mark only one oval.

○ Employed full time
○ Employed Part time
○ Student
○ Retired
○ Unemployed
○ Other:

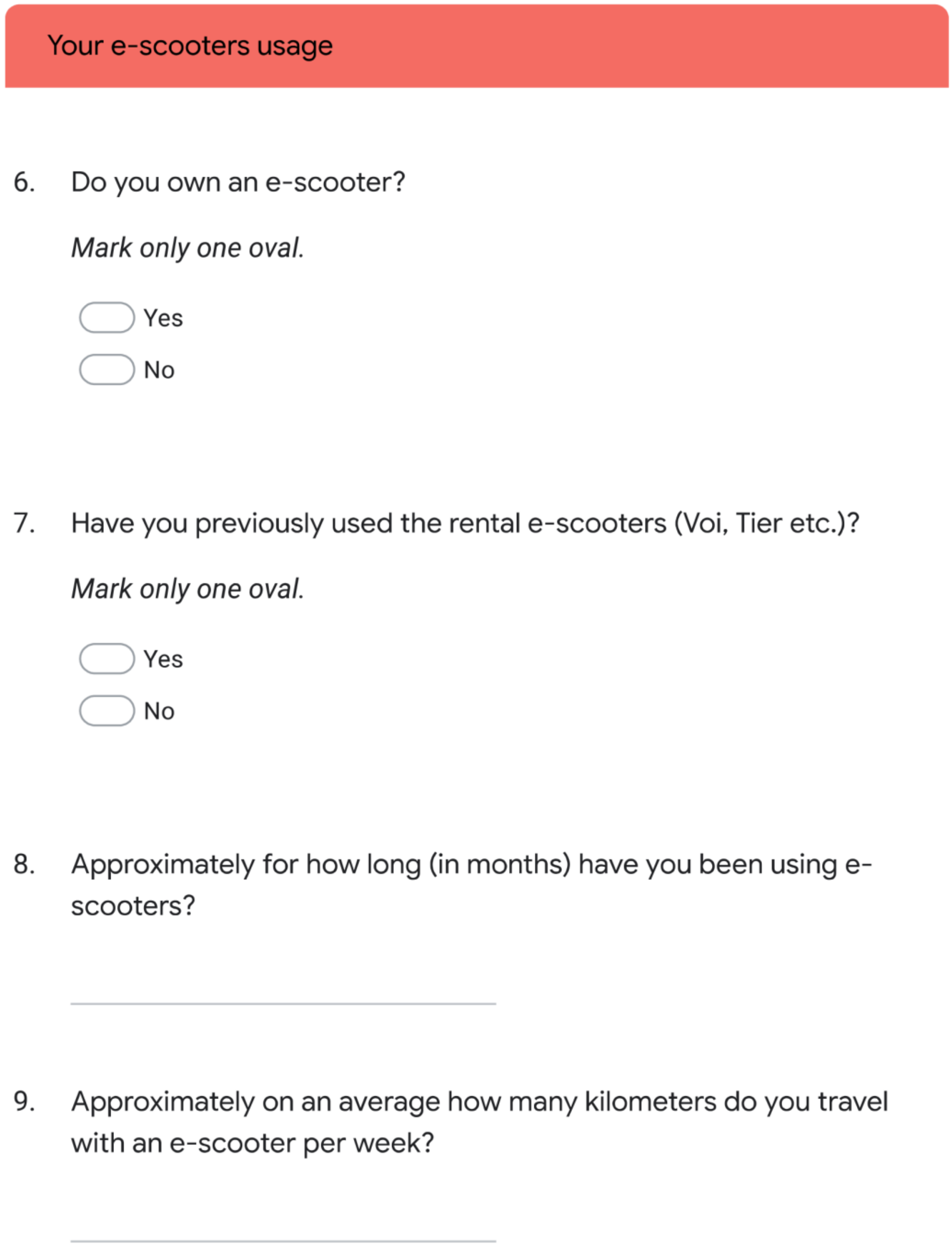


5. How often do you use following means of transportation?

This is on an average throughout the year irrespective of the seasons (Summer, Winter etc.).

Mark only one oval per row.

	Every day	4-5 days per week	2-3 days per week	1 day per week	< 1 Day per week	Never
Foot	○	○	○	○	○	○
Bicycle	○	○	○	○	○	○
Electric Bicycle	○	○	○	○	○	○
E-scooter	○	○	○	○	○	○
Car (As driver)	○	○	○	○	○	○
Car(As passenger)	○	○	○	○	○	○
Bus	○	○	○	○	○	○
Tram	○	○	○	○	○	○
Ferry	○	○	○	○	○	○
Other Micro-Mobility Vehicles (E-Monowheel or E-Skateboard etc.)	○	○	○	○	○	○



Your e-scooters usage

6. Do you own an e-scooter?

Mark only one oval.

○ Yes
○ No

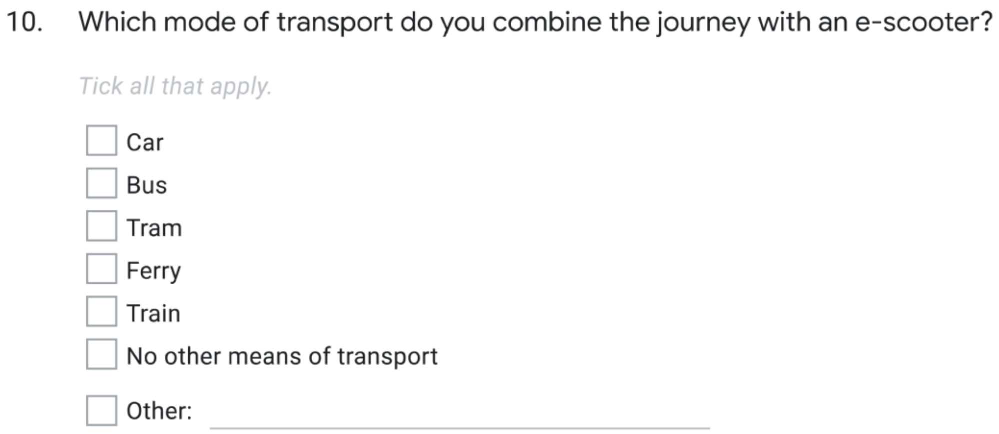
7. Have you previously used the rental e-scooters (Voi, Tier etc.)?

Mark only one oval.

○ Yes
○ No

8. Approximately for how long (in months) have you been using e-scooters?

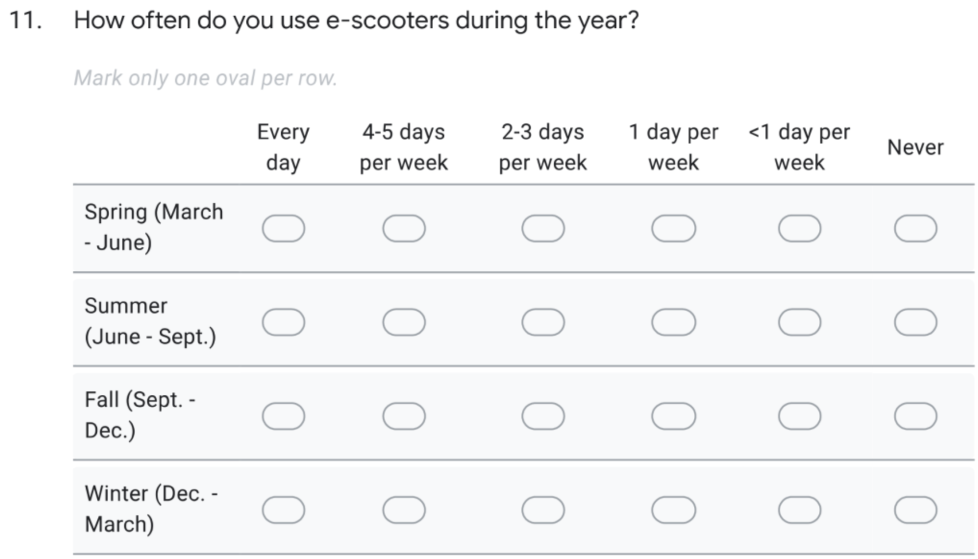
9. Approximately on an average how many kilometers do you travel with an e-scooter per week?



10. Which mode of transport do you combine the journey with an e-scooter?

Tick all that apply.

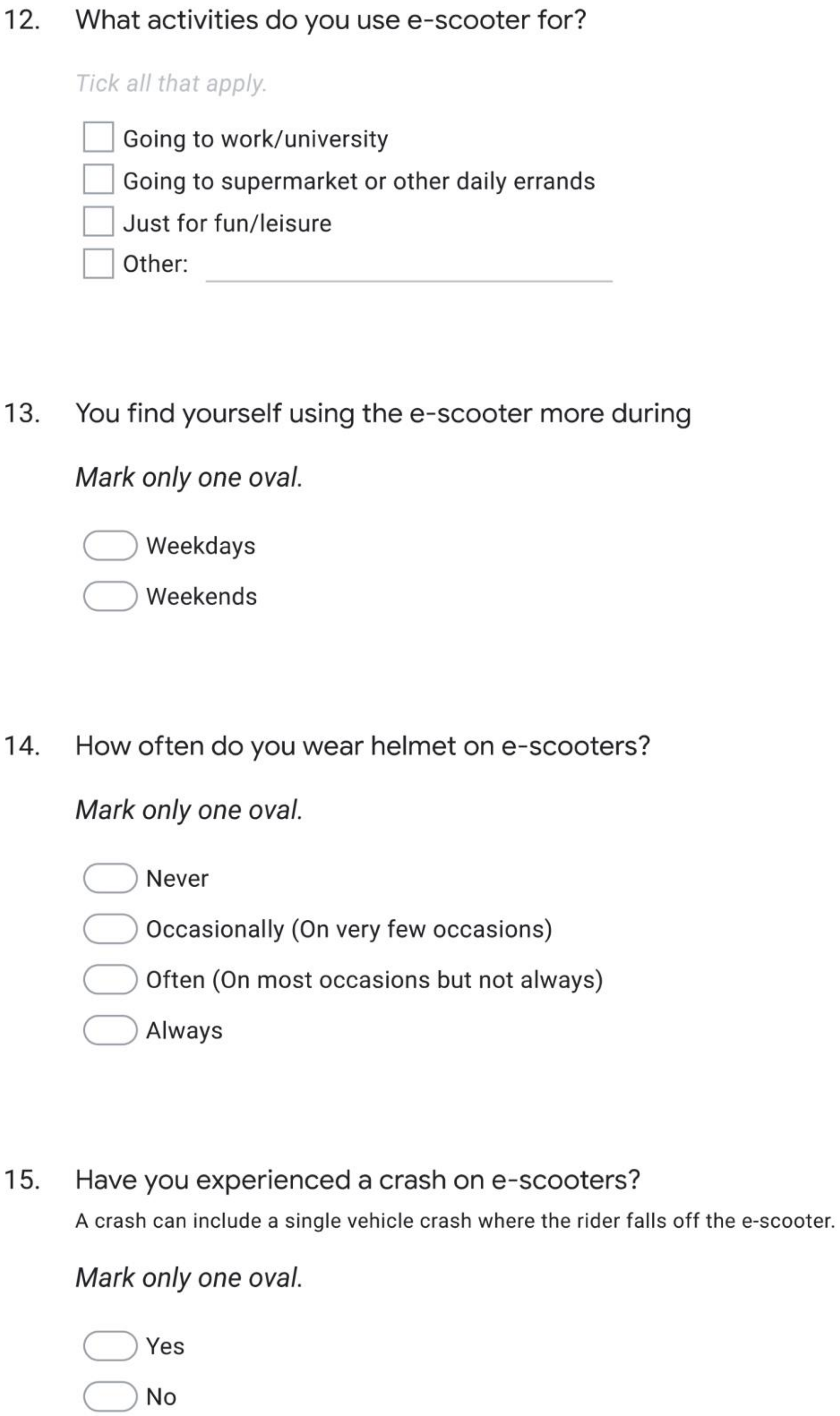
☐ Car
☐ Bus
☐ Tram
☐ Ferry
☐ Train
☐ No other means of transport
☐ Other: ____________________



11. How often do you use e-scooters during the year?

Mark only one oval per row.

	Every day	4-5 days per week	2-3 days per week	1 day per week	<1 day per week	Never
Spring (March - June)	○	○	○	○	○	○
Summer (June - Sept.)	○	○	○	○	○	○
Fall (Sept. - Dec.)	○	○	○	○	○	○
Winter (Dec. - March)	○	○	○	○	○	○



12. What activities do you use e-scooter for?

Tick all that apply.

☐ Going to work/university
☐ Going to supermarket or other daily errands
☐ Just for fun/leisure
☐ Other: ____________________

13. You find yourself using the e-scooter more during

Mark only one oval.

○ Weekdays
○ Weekends

14. How often do you wear helmet on e-scooters?

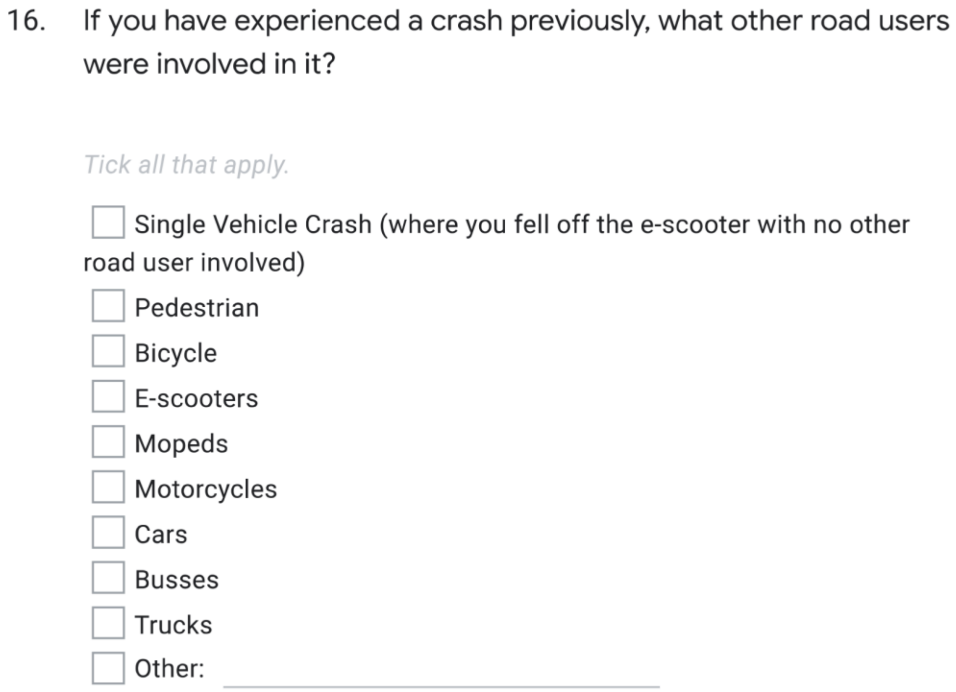
Mark only one oval.

○ Never
○ Occasionally (On very few occasions)
○ Often (On most occasions but not always)
○ Always

15. Have you experienced a crash on e-scooters?
A crash can include a single vehicle crash where the rider falls off the e-scooter.

Mark only one oval.

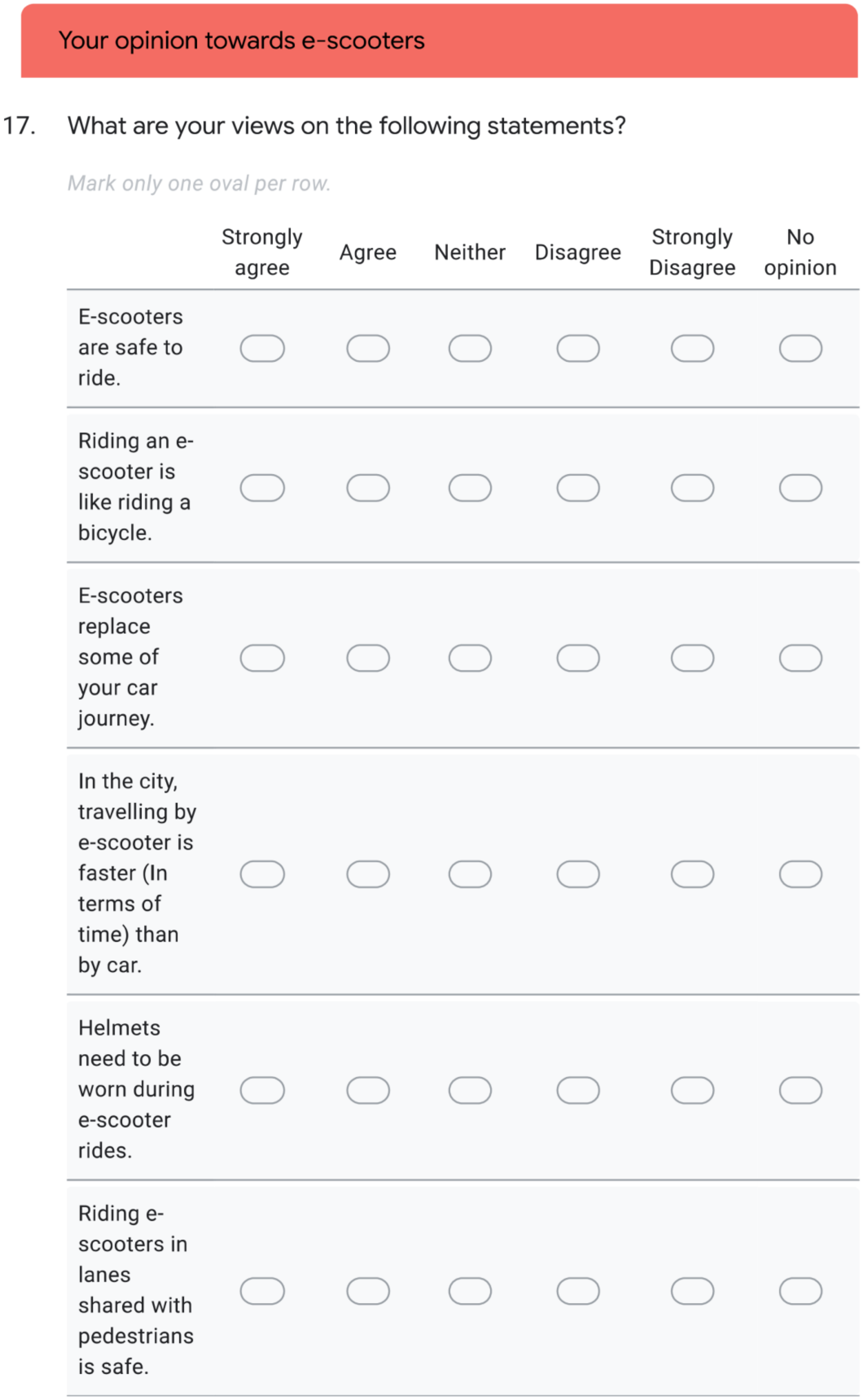
○ Yes
○ No



16. If you have experienced a crash previously, what other road users were involved in it?

Tick all that apply.

☐ Single Vehicle Crash (where you fell off the e-scooter with no other road user involved)
☐ Pedestrian
☐ Bicycle
☐ E-scooters
☐ Mopeds
☐ Motorcycles
☐ Cars
☐ Busses
☐ Trucks
☐ Other: ____________________

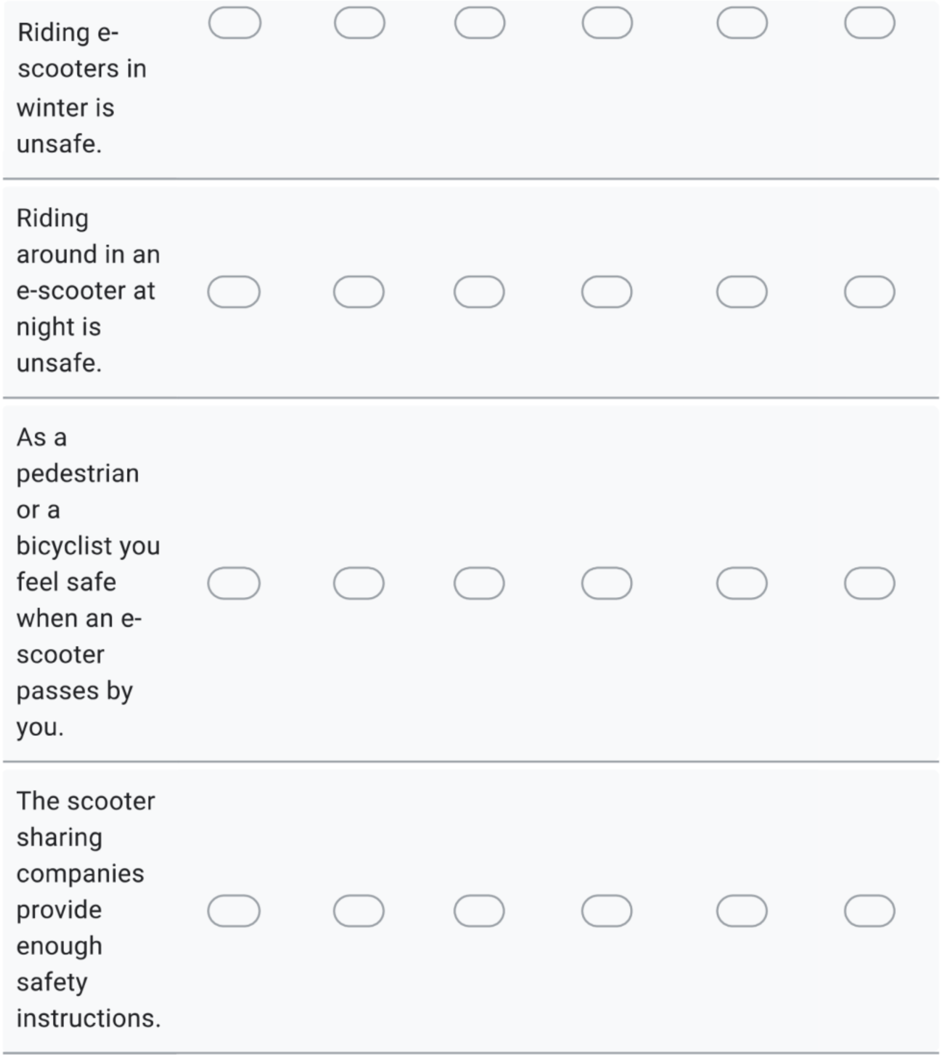


Your opinion towards e-scooters

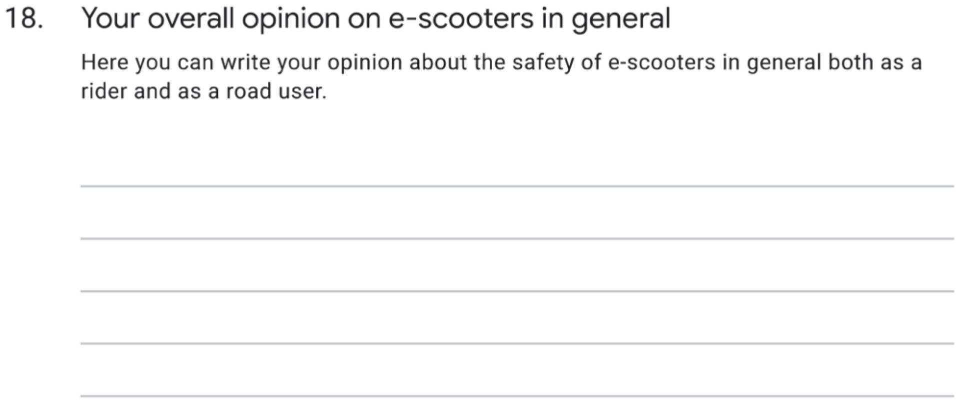
17. What are your views on the following statements?

Mark only one oval per row.

	Strongly agree	Agree	Neither	Disagree	Strongly Disagree	No opinion
E-scooters are safe to ride.	○	○	○	○	○	○
Riding an e-scooter is like riding a bicycle.	○	○	○	○	○	○
E-scooters replace some of your car journey.	○	○	○	○	○	○
In the city, travelling by e-scooter is faster (In terms of time) than by car.	○	○	○	○	○	○
Helmets need to be worn during e-scooter rides.	○	○	○	○	○	○
Riding e-scooters in lanes shared with pedestrians is safe.	○	○	○	○	○	○



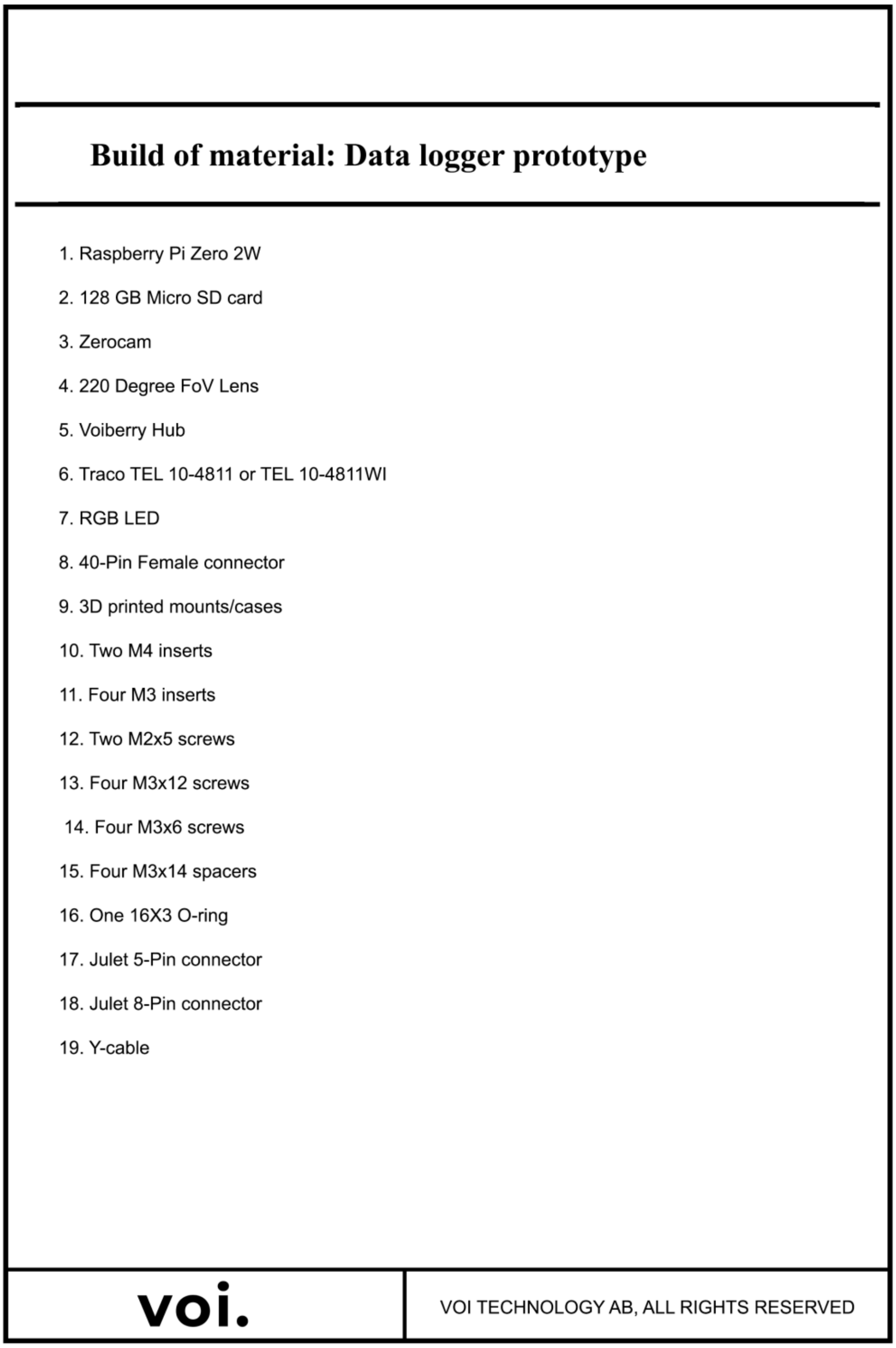
Riding e-scooters in winter is unsafe.	○	○	○	○	○	○
Riding around in an e-scooter at night is unsafe.	○	○	○	○	○	○
As a pedestrian or a bicyclist you feel safe when an e-scooter passes by you.	○	○	○	○	○	○
The scooter sharing companies provide enough safety instructions.	○	○	○	○	○	○



18. Your overall opinion on e-scooters in general

Here you can write your opinion about the safety of e-scooters in general both as a rider and as a road user.

C. Build of materials



Build of material: Data logger prototype

1. Raspberry Pi Zero 2W
2. 128 GB Micro SD card
3. Zerocam
4. 220 Degree FoV Lens
5. Voiberry Hub
6. Traco TEL 10-4811 or TEL 10-4811WI
7. RGB LED
8. 40-Pin Female connector
9. 3D printed mounts/cases
10. Two M4 inserts
11. Four M3 inserts
12. Two M2x5 screws
13. Four M3x12 screws
14. Four M3x6 screws
15. Four M3x14 spacers
16. One 16X3 O-ring
17. Julet 5-Pin connector
18. Julet 8-Pin connector
19. Y-cable

voi.

VOI TECHNOLOGY AB, ALL RIGHTS RESERVED

DEPARTMENT OF MECHANICS AND
MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY