Article

Advancing Sustainable Mobility: A Data Acquisition System for Light Vehicles and Active Mobility

Matteo Verzeroli [ID], Luigi Gaioni *[ID], Andrea Galliani [ID], Luca Ghislotti [ID], Paolo Lazzaroni [ID] and Valerio Re [ID]

Department of Engineering and Applied Sciences, University of Bergamo, 24129 Bergamo, Italy;
matteo.verzeroli@unibg.it (M.V.); andrea.galliani@unibg.it (A.G.); luca.ghislotti@unibg.it (L.G.);
paolo.lazzaroni@unibg.it (P.L.); valerio.re@unibg.it (V.R.)
* Correspondence: luigi.gaioni@unibg.it

Abstract: Active mobility and light vehicles, such as e-bikes, are gaining increasing attention as sustainable transportation alternatives to internal combustion solutions. In this context, collecting comprehensive data on environmental conditions, vehicle performance, and user interaction is crucial for improving system efficiency and user experience. This paper presents a data acquisition system designed to collect data from multiple sensor platforms. The architecture is optimized to maintain low power consumption and operate within limited computational resources, making it suitable for real-time data acquisition on light vehicles. To achieve this, a data acquisition module was developed using a single-board computer integrated with a custom shield, which also captures data related to the assistance of an e-bike motor through a wireless interface. The paper provides an in-depth discussion of the architecture and software development, along with a detailed overview of the sensors used. A demonstrator was created to verify the system architecture idea and prove the potentialities of the system overall. The demonstrator has been qualified by professional and semi-professional riders in the framework of the Giro-E, a cyclist event which took place in May 2024, on the same roads of the Giro d'Italia. Finally, some preliminary analyses on the data acquired are provided to show the performance of the system, particularly in reconstructing the user behavior, the environmental parameters, and the type of road.

Keywords: light mobility; low power; software architecture; air quality index; MOx sensors

1. Introduction

The MOST project (Centro Nazionale per la Mobilità Sostenibile) is a key initiative funded under the National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza-PNRR), part of the broader Next Generation EU program. This ambitious project aims to advance research across different domains of sustainable mobility, including air, road, rail, and water transportation, along with light vehicles and active means of transport. The project brings together a consortium of Italian universities, research institutes, and private companies, structured into 14 Spokes, each dedicated to a specific area of research. With a focus on light vehicles and active mobility, Spoke 5 is led by the University of Bergamo.

Active mobility refers to means of transport such as walking and cycling, either as standalone options or associated with public transportation. Such an approach to urban mobility offers a number of benefits, including improvements in urban planning, transportation efficiency, and public health, by promoting physical activity among the population [1]. In the last few years, the proliferation of electric vehicles, such as e-bikes and e-mopeds, has significantly contributed to the growth of active multimodal transportation. Moreover, increasing public awareness of the environmental impact of emissions from vehicles equipped with internal combustion engines is promoting a shift toward sustainable mobility options, including light vehicles. By reducing car dependency, such vehicles are

expected to mitigate traffic congestion in urban areas [2]. However, to ensure safety, local authorities must implement appropriate regulations and safe lanes, as light vehicle users are among the most vulnerable road users [3]. Additionally, the rise in shared mobility services, such as bike and scooter-sharing programs, is further enhancing the accessibility and convenience of active and sustainable transportation options, contributing to a more flexible urban mobility ecosystem [4].

A key area of investigation within the MOST project is the application of advanced technologies, typically associated with electric mobility, to light electric vehicles. This includes the integration of sensor networks, Internet of Things (IoT) devices, and vehicle-to-everything (V2X) communication infrastructures. The transport industry is increasingly leveraging big data to enhance safety, improve vehicle performance, and support sustainable growth [5]. Miniaturization and reduced power consumption of sensors and processing units, coupled with advancements in V2X and IoT communication, promote real-time, high-resolution environmental data collection in smart city applications. These data can include information on air quality, noise pollution, and road surface conditions, providing valuable insights for local authorities to address urban issues effectively. Visualizing such a big amount of diversified data to support human decision-making is still a field of research [6]. Due to limitations in battery capacity and computational power in light vehicles compared to their electric vehicle counterparts, achieving reliable data collection requires the development of low-power data acquisition systems. With this challenge in mind, the MOST project has established a dedicated work package (i.e., the WP4.1) aimed at advancing key enabling technologies, including sensors, ICT infrastructures and wearable modules, which will be crucial for gathering reliable data on vehicles, users, and the surrounding environment.

A number of data acquisition systems, specifically developed for smart mobility, and bikes in particular, are available and described in the literature, [7–9]. These systems may be exploited in such a way as to gather valuable data for different applications, including, but not limited to, urban planning, safety analysis, and user experience. Data acquisition systems share several key components, such as sensors (collecting data related to different aspects of the bike environment and operation), communication modules, and blocks that provide a suitable power supply for the proper operation of the system. While data acquisition systems specifically developed for e-bikes and light vehicles offer a number of benefits, there are also challenges to consider, such as data privacy and security, battery life, integration with existing infrastructure, and the need for advanced analytics to extract meaningful insights from collected data.

In [7], the authors describe a real-time, open-source, and open-hardware platform designed to collect data from e-bikes. The system monitors location (through a global positioning system), rider control data (level of assistance), and other custom sensor inputs. The system, called SEMS (smart e-bike monitoring system), provides readings to an online interface for data analysis, which can be useful for riders to check their data and share them on social media.

In [8], an IoT-based system for real-time air quality monitoring and for enhancing rider safety is proposed. PM2.5 data are collected and transmitted to a cloud platform for analysis. The authors developed a mobile application which informs users about air quality along their routes. Moreover, the developed device features ultrasonic sensors providing collision avoidance warnings via visual and auditory alerts on the user's smartphone.

In [9], the authors present a prototype of a smart bike ecosystem designed for urban air quality monitoring. A system was developed to collect and aggregate data on PM and environmental conditions, including air pressure, temperature, and humidity. Preliminary testing successfully demonstrated the feasibility of leveraging a network of bikes to gather real-time air quality information.

The paper is organized as follows: Section 2 describes the overall system architecture being developed in the framework of the WP4.1 of the MOST project, which includes a main acquisition system gathering data from a number of modules, intended for the monitoring of the vehicle, the environment, and the user. Section 3 provides the reader with a detailed description of the main data acquisition module, particularly focusing on the software architecture, while Section 4 discusses preliminary test results as obtained from the first fabricated samples of the demonstrator.

2. System Architecture

The architecture of the system being developed in the framework of WP4.1 is shown in Figure 1. The system is designed to monitor vehicle performance, the environment, and rider's condition, and it leverages several key modules, each serving a specific purpose [10].
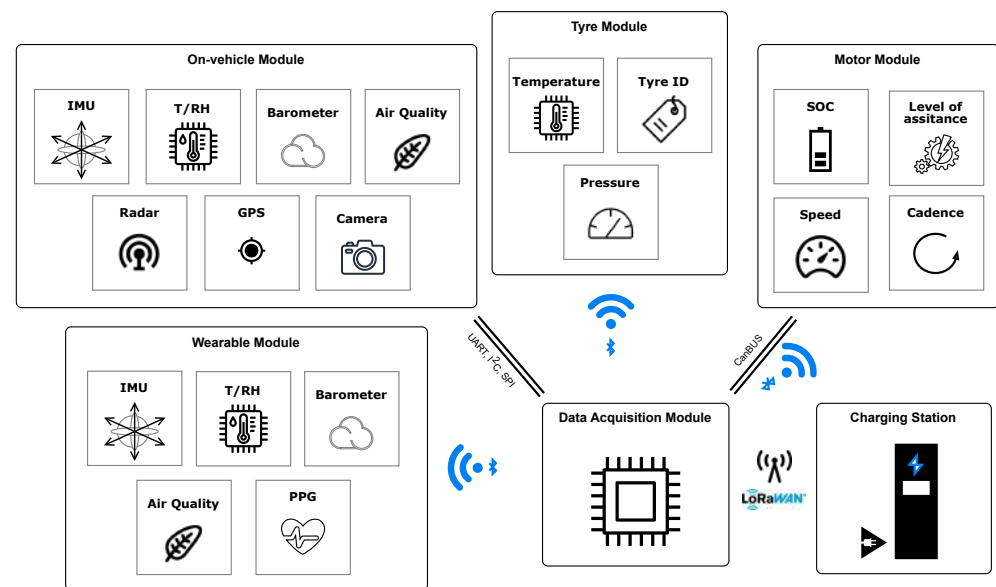

Figure 1. System architecture envisioned for the final setup.

The system is conceived for light vehicles such as e-bikes or scooters. The developed demonstrator integrates 3D-printed hooks specifically designed for e-bike handlebars, while the power supply for the electronics is provided by the e-bike battery pack. A different 3D-printed design can be used to adapt the system to other light vehicles. A standard, external battery (providing a supply voltage higher than 6 V) may be connected to the system if the light vehicle is not equipped with a power supply unit.

The system includes a data acquisition (DAQ) module conceived to aggregate the vehicle, the environment, and the user data and to monitor the vehicle through their elaboration. The system consists of six key components: the data acquisition module (DAM), the wearable module (WM), the on-vehicle module (OM), the tire module (TM), the motor module (MM), and the charging station (CS). The main component is the DAM, which serves as the system coordinator. Its primary function is to gather data from the different interconnected modules through multiple communication protocols, both wired and wireless. The DAM is also responsible for handling data storage, either saving it locally or transmitting aggregated datasets to external devices for further analysis. In contrast to traditional DAQ systems for light vehicles, such as the one outlined in [11], the DAM merges all gathered measurements into a single file, in such a way that post-processing is not needed to combine separate data streams. A comprehensive description of the DAM is provided in Section 3.

A smart wearable device will be utilized for user monitoring. Specifically, the device can be a smart bracelet, integrating a number of sensors for physiological and environmental

data collection, along with wireless communication capabilities, such as Bluetooth Low Energy (BLE). Given that the module is intended to operate on a compact battery, implying a limited capacity, optimizing power efficiency is of crucial to ensure prolonged operation without the necessity of recharging, thereby enhancing user convenience and overall system performance. An inertial measurement unit (IMU) can be employed to monitor user movements and detect vibrations, thus providing a set of data on the dynamic behavior of a cyclist. At the same time, a photoplethysmography (PPG) module can be integrated to continuously assess the rider's physiological state. Through the analysis of PPG signals, critical health metrics, such as heart rate and blood oxygen saturation (SpO_2), can be accurately measured, offering real-time insights into cardiovascular performance. For environmental monitoring, different sensors can be used to measure temperature, humidity, atmospheric pressure, air quality, and other variables. These sensors contribute to building a comprehensive understanding of the rider's surroundings, ensuring safety and comfort during outdoor activities. Furthermore, to enhance air quality monitoring, the use of low-power, miniaturized metal oxide (MOx) sensors has been proposed for outdoor applications. However, additional research is needed to assess the performance, accuracy, and energy efficiency of such sensors, particularly in outdoor environments, where factors such as varying pollutant levels and weather conditions may influence sensor reliability.

The OM is interfaced with the central processing unit through a wired communication protocol, such as the Universal Asynchronous Receiver-Transmitter (UART), Inter-Integrated Circuit (I^2C), or Serial Peripheral Interface (SPI), to ensure higher data acquisition rates and reliable, real-time data transmission. This module can be equipped with an IMU that combines an accelerometer and a gyroscope, enabling precise monitoring of vehicle dynamics, vibration tracking, fall detection, and a detailed assessment of driving patterns and style. A GPS (Global Positioning System) module will be integrated to enable precise tracking of the vehicle's position and speed. This will not only provide real-time data on vehicle movement but also establish an absolute spatial reference for all other collected data, enabling more accurate analysis. With regard to the environmental monitoring functions of the OM, sensors for temperature, humidity, pressure, and air quality will be employed to continuously assess the surrounding environmental conditions. The wearable module will include the same types of sensors. Moreover, to enhance the safety of both the rider and other road users, advanced sensing technologies, such as radar, ultrasound, and cameras, can be exploited to enable obstacle detection. These sensors contribute to collision avoidance by identifying potential hazards in the vehicle's path. In contrast to the wearable module, which is constrained by a limited power budget, the OM can utilize the e-bike battery pack as its main power source. This significantly reduces power consumption concerns, allowing for the integration of more energy-demanding components without compromising performance or battery life.

The status of the vehicle is continuously monitored through the TM and the MM. The TM is designed to provide key information, such as the tire identification number, temperature and pressure. These data are valuable for assessing the tire maintenance condition, detecting potential punctures, and identifying issues that may affect performance or safety. Meanwhile, the MM is responsible for retrieving data from the e-bike control unit. Within the scope of the MOST project, the manufacturer will offer open access to different parameters, including battery status, assistance level, current speed, and cadence. These pieces of information are essential for analyzing correlations between different variables, such as the relationship between the rider's heart rate and the level of motor assistance or the impact of road conditions on battery discharge rates. By examining these factors, a more comprehensive understanding of vehicle performance, energy efficiency, and user interaction can be achieved.

The DAM will integrate a LoRaWAN (Long Range Wide Area Network) module to establish connectivity with fixed charging stations. These charging stations will serve as gateways for transmitting aggregated data to cloud-based platforms, where the data will undergo further analysis and processing. Although LoRa technology offers limited

up-link bandwidth, which restricts the volume of data that can be transmitted in a single instance, it enables low-power wide area network (LPWAN) connectivity. This characteristic makes LoRaWAN particularly well suited for energy-efficient, long-range communication, allowing the DAM to maintain low power consumption while still supporting reliable data transfer over extended distances, particularly in scenarios where continuous, real-time data streaming is not required.

As the usage of light vehicles in cities has increased over time (e.g., with food-delivery riders, rental e-bikes, $\cdots$), the possibility of monitoring the overall state of vehicles and user behavior in an urban context became of interest to bike manufacturers, rental agencies, and other stakeholders in the business, that can benefit either directly through bike sales and rentals, or indirectly by tracking riders during their work shifts. The advent of smart cities also makes a low-power, modular, easily integrable, IoT-ready system such as the one proposed worthy of notice. The integration of a LoRaWAN module will make the system ready to exchange data in a connected, data-driven environment. It will also enable local aggregators of data (such as recharge stations, traffic lights, and so on) to infer information on the urban setting evolution (road conditions, state of the traffic, $\cdots$), and the behavior of the citizens (complying to the traffic light signals, crashes, $\cdots$) in a qualitative way, while maintaining the privacy of the users.

3. Data Acquisition Module

The DAM is a key component of the architecture, serving as the central management unit for the whole system. It is responsible for collecting data from the other modules, supervising their operation, as well as running algorithms to analyze the gathered data and provide real-time feedback on the status of the bike, the user, and the environment. The module is conceived as a general-purpose processing unit with an ad hoc software application installed to provide the aforementioned features. The hardware and software must be kept independent from the composition of the other modules, leaving the possibility to extend or modify the system architecture with limited effort. In addition, the DAM must support both wired and wireless communication protocols to support the different module interfaces. Single-board computers (SBCs), such as Raspberry Pi boards, are off-the-shelf, low-cost solutions suitable for this application. They ensure sufficient computational power while maintaining small dimensions and low power consumption. Due to a general purpose input/output (GPIO), SBCs support different wired communication protocols, including UART, I^{2}C, and Serial Peripheral Interface (SPI). The UART protocol is often used to exchange data between a microcontroller and peripherals, while I^{2}C and SPI are typically used to interact with sensors. Moreover, recent versions of Raspberry Pi platforms include wireless connectivity (e.g., Bluetooth, BLE and Wi-Fi), together with an Ethernet connection port. They also provide four USB connectors and a camera serial interface (CSI) to handle the interface with a camera. The power consumption of these boards typically does not exceed 10 W. A Raspberry Pi 3B+ has been chosen to deploy the DAM because it supports all communication interfaces of interest and has sufficient computational power. All the peripherals are controlled by Raspberry Pi OS Lite, a Unix-like operating system based on the GNU/Linux distribution for the Raspberry Pi family. The system can be accessed through a Secure Shell (SSH) protocol to interact with the OS. Using a general-purpose OS, it is possible to build platform-independent applications which can be made portable to different device architectures.

A key element of theDAM is the software application deployed on the Raspberry Pi board, which has to interact with the modules described in Section 2. Since the modules operate as independent entities, several challenges must be addressed to ensure the accuracy of data acquisition. The diversified communication protocols and different sample rates of each acquisition module are managed by the use of concurrent processes running on the processor to exploit all the computational power available. The first step in designing the software application is the requirements analysis. The system must ensure the composability and extendibility of the system, allowing the integration of different modules

with minimal effort while guaranteeing the portability and reusability of the system across various host platforms. The module has to acquire the data reliably, minimizing data loss. Since the Raspberry Pi boards have a multi-core processor, the application is developed as a multi-process program to exploit all the computational power and allow for concurrent tasks to be executed simultaneously. The support for continuous integration and continuous delivery (CI/CD) practices is useful since it facilitates ongoing improvements and ensures reliable updates. The software development life cycle (SDLC) selected is the incremental approach, where various phases—requirements definition, design, implementation, coding, testing, and deploying—are repeated over time to build the whole application. Moreover, applying the best practises of component-based software engineering (CBSE) ensures the quality attributes of composability and extensibility. The pivotal element of the CBSE is the software component, which is a modular, deployable, and replaceable part of a system that hides implementation and exposes a set of interfaces [12]. The loose coupling property of each component improves the maintainability of the system as well as its reusability, composability, and flexibility. The advantage of using a component-based architecture combined with an incremental development approach is the ability to gradually assemble the final application. Figure 2 depicts a UML-like component diagram of the proposed architecture.
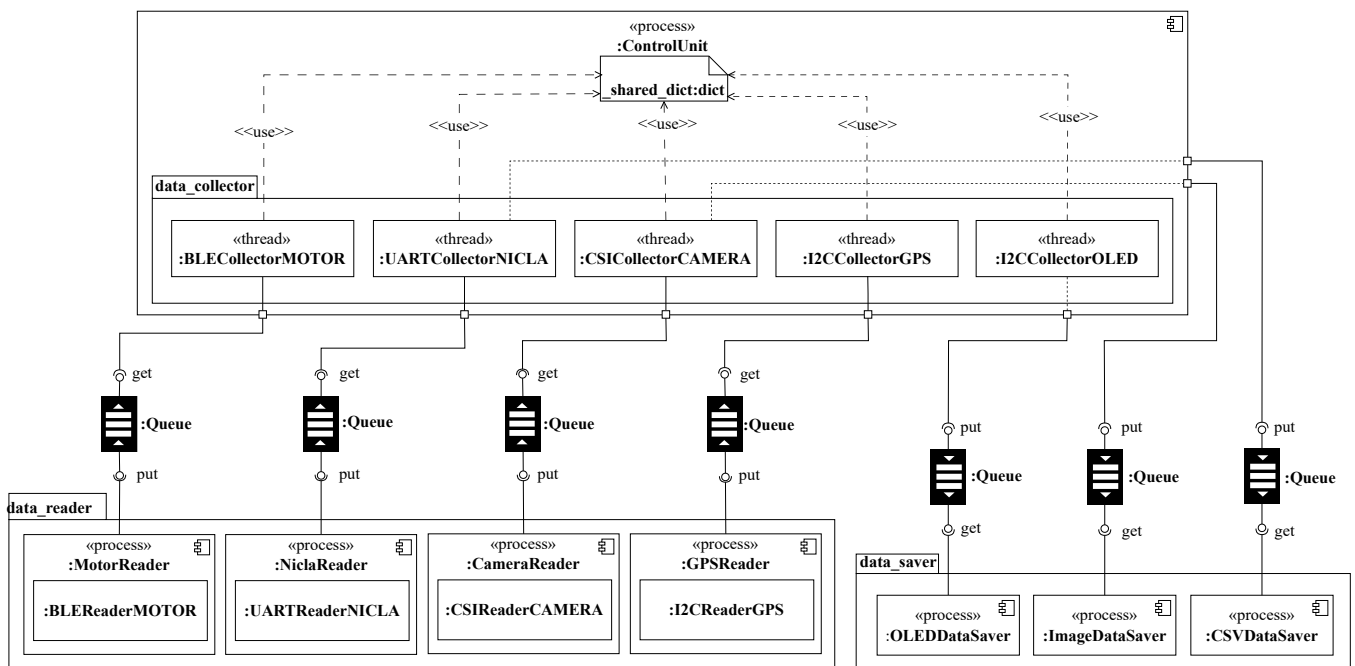
Figure 2. UML-like representation of the software architecture of the data acquisition module.

The application is organized into three layers: a data reader layer, a data collector layer, and a data saver layer. Multi-process queues are used as connectors between layers due to their operational behavior that are now described. A first-in-first-out (FIFO) queue can guarantee asynchronous communication, acting as a buffer and making it possible to decouple the sampling rate of the components involved. In addition, a queue is a mediator preventing the process from having access to the internal state of the other processes, thereby preserving the independence of the modules. Additionally, it makes the system more fault-tolerant since if a process fails or is temporarily unavailable, it does not affect the execution of another one. Moreover, a queue can store data until the failed process is restored. Each layer has a specific responsibility to ensure correct data flow and efficient management of the module. The data reader layer manages the communication with the modules connected to the DAM, and its main purpose is to retrieve raw data from the sensor platforms. Each connected board is related to its own data reader, where the

communication protocol is implemented. The unprocessed data are collected by the data collector layer, which interprets raw bytes as packed binary types. It is also the entry point of the whole application, which creates all the software entities and initializes all the components. A data structure, shared by all the data collectors, aggregates all the data processed. It summarizes the latest status of the entire system, which is then sent to the data saver layer to save the information. The data saver layer is tasked with storing the data on a permanent storage medium (e.g., micro SD cards, USB flash drive) and providing an interface with output modules. These modules are connected to output devices, such as display or IoT communication modules, which send aggregated data to external devices. Some typical IoT communication protocols are Lora and MQTT, which can provide a connection with the cloud for further data processing.

The application designed for the DAM is developed using the Python programming language. Although Python is less efficient than other programming languages in terms of memory management and processing speed, it has some distinct advantages. It speeds up development due to the rich ecosystem of open-source libraries, making it possible to prototype quickly even with volatile requirements. Additionally, it features cross-platform compatibility and extensibility with C for performance-demanding tasks. Moreover, there is a Python package that helps to write multi-processing applications. As depicted in Figure 2, each component of the data reader layer and the data saver layer is deployed as a process created by the OS. The use of processes not only enables to run code on parallel on multi-core CPU architectures but also fosters the independence of the components because each process is a program in execution with its own address space, that is, a list of memory locations where the process can read and write [13]. By contrast, the data collector modules are implemented as threads, an execution path of the main control unit process, and they have an address space in common. As a result, they can share the data structure, which contains all the latest status of the system, as noted earlier. However, there is a key difference between processes and threads in Python applications. The Global Interpreter Lock (GIL), a mechanism used by the CPython interpreter, prevents threads of the same process from being executed simultaneously, simplifying their implementation and protecting them from concurrent access. Only processes can be executed in parallel on multi-core CPUs because they are separate instances of the Python interpreter, whereas threads compete for execution. The Python multi-processing and threading module supports the development of multi-process applications. They feature inter-process communication (IPC) connectors (e.g., queues and shared memory), allowing different processes to share data while maintaining process isolation. These software elements are overseen by a process manager task which handles the resources shared among the processes. Configuration files are used to set user-configurable parameters, such as sample rates, timeouts, and ports of connection to accommodate changes in requirements. The general workflow of the DAM application is as follows: A sensor module acquires data and transmits them on a communication channel to the DAM. A dedicated component of the data reader layer reads the raw data and sends them to the corresponding data collector through a queue. The data collector processes data and updates the shared data structure, which contains the current status of the system with the new data and a timestamp. Finally, based on the selected configuration, the shared dictionary can be sent, using an inter-process queue, to the data saver layer to be saved.

Following the description of the hardware and software implementation of the DAM, the developed system also has two other modules: the OM and the MM. The OM gathers data about the vehicle and the environment through several on-board sensors. The module is a Raspberry Pi shield connected directly to the DAM module via the GPIO, which serves as an interface to the communication peripherals of the Raspberry Pi board and contributes to the mechanical support of the module. A 3D representation of the final setup is shown in Figure 3.
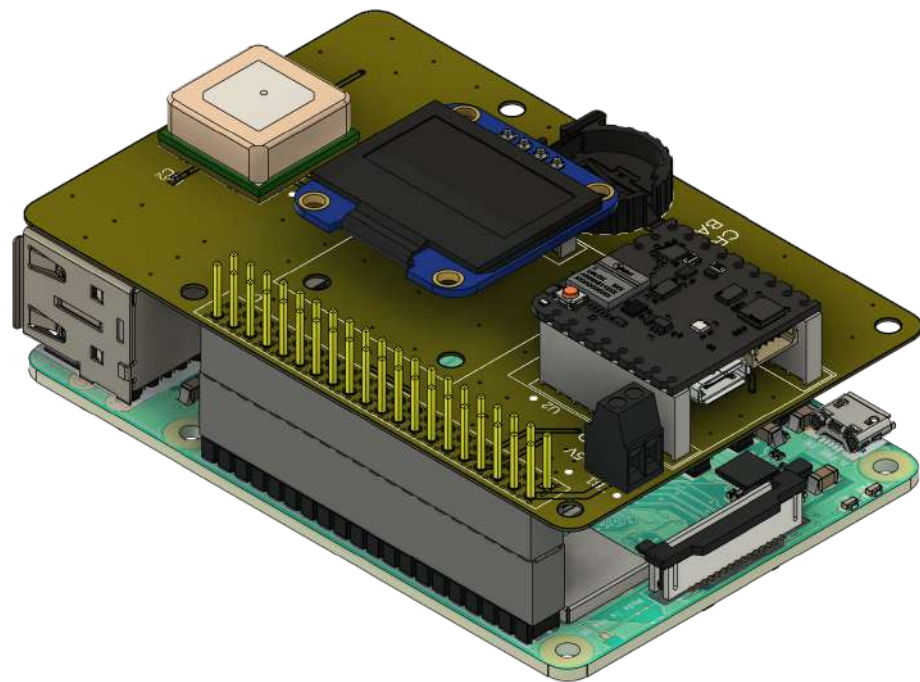
Figure 3. 3D representation of the developed system.

An IMU and environmental sensors, including temperature, humidity, pressure, and metal oxide air-quality sensors, are provided by the Nicla Sense ME, a small form factor, low-power board with industrial-grade sensors [14]. The Nicla Sense ME is a compact development board designed by Arduino Pro in collaboration with Bosch Sensortec. It integrates cutting-edge sensors and offers a simple programming structure. This board is ideal for creating smart sensing applications in the IoT domain, as it has a powerful processor that efficiently handles data processing tasks. This efficiency translates to low power consumption, making the board ideal for projects that rely on battery power. This paves the way for creating long-lasting wearable devices or portable environmental monitoring systems. The core of the Nicla Sense ME consists of its four integrated Bosch Sensortec sensors, described in the following. The BHI260AP is an AI smart sensor that combines a six-axis IMU with a 32-bit programmable microcontroller and different software functionalities. The six-axis IMU includes a three-axis accelerometer and a three-axis gyroscope, allowing precise motion sensing and orientation tracking. The BHI260AP is designed to be ultra-low power so that it can be used in devices that need to last a long time on a battery, such as fitness trackers and wearables. As reported in the device datasheet [15], the BHI260AP is mainly intended to be used as a co-processor offloading the main CPU from any sensor data processing-related tasks, including sensor fusion, data batching, position tracking, activity recognition and gesture detection, with high precision and low latency while significantly reducing overall system power consumption. When used as a co-processor, the BHI260AP supports a wide variety of host CPU devices, ranging from small MCUs to multi-core application processors. The BHI260AP communicates with the host CPU through its primary high-speed I2C or SPI interface. The second sensor integrated into the Nicla Sense Board is the BMM150 magnetometer, a compact sensor that can detect magnetic fields in three directions (X, Y, and Z). It is particularly well suited for use in electronic compasses and navigation systems due to its low power consumption, low noise (which allows for accurate magnetic field measurements), and small size. The sensor has a measurement range of $\pm 1300\,\mu\text{T}$ on the X- and Y-axes and $\pm 2500\,\mu\text{T}$ on the Z-axis, which is beneficial for applications requiring good sensitivity in the vertical direction [15]. Its resolution is $0.3\,\mu\text{T}$, providing detailed magnetic field information. The sensor supports standard I2C and SPI communication interfaces, making integration with various microcontrollers easy. It

operates at both 3.3 V and 5 V and can function in environments with temperatures ranging from −40 °C to 85 °C [15]. The data output rate reaches 10Hz in normal mode, allowing for real-time magnetic field measurements. The third main component of the Nicla Sense ME is the BMP390 pressure sensor. This is a low-power (typically around 3.2 µA at 1 Hz), which is optimal for battery-powered applications, low-noise, 24-bit absolute barometric pressure sensor developed by Bosch Sensortec. This digital, high-performance sensor is ideally suited for a wide range of altitude tracking applications. The BMP390 operates in the pressure range of 300 to 1250 hPa, with a supply voltage of 1.2 V to 3.6 V. Its compact package dimensions ($2.0 \times 2.0 \times 0.75\,\mathrm{m}^3$) make it an excellent choice for space-constrained applications. It offers high-precision pressure and temperature measurement with an absolute pressure accuracy of $\pm 0.50\,\mathrm{hPa}$ and temperature accuracy of $\pm 1.5\,°C$ [16]. The fourth component of the Nicla Sense ME is the BME688 sensor, a four-in-one environmental sensor that integrates temperature and humidity sensors, a barometer, and a MOx air quality sensor in a small $3.0 \times 3.0 \times 0.93\,\mathrm{m}^3$ package. The Bosch Sensortec Environmental Cluster (BSEC), a close-source library offered by Bosch Sensortec, features high-level signal processing and a fusion algorithm to easily calculate environmental parameters, such as an air quality index, estimated CO_2 concentration, breath volatile organic compounds (VOCs) levels, and classification models [17].

Since a sampling rate of 100 Hz is required in this application, the choice was made to stream the data to the DAM through a UART interface instead of using the on-board Bluetooth module. Custom firmware was developed to implement the communication protocol with the DAM. A ticker triggers the data transfer every 10 milliseconds through the UART interface. The reliability of the data transfer is checked through an eight-bit cyclic redundancy check (CRC-8), which ensures the correctness of the transmission. Moreover, each data packet is encoded using consistent overhead byte stuffing (COBS), a technique, which helps to relax the dependency of the communication entities on the data packet length. A SAM-M8Q GPS module provides the reference in space and time of the acquired data to the entire system. The DAM directly reads the information about the current latitude, longitude, speed, altitude, and time from the module through an I²C link. The GPS module uses a CR1220 backup battery to maintain the valid time and GNSS orbit data when the board is powered off [18]. This information improves the GNSS performance at start-up. Finally, an OLED display is used to give feedback to the rider. It is controlled by a specific component of the data saver layer via an I²C interface. A Raspberry Pi camera module finalizes the OM and acquires images about the environment. Due to the development of intelligent algorithms, these photos can help to perform certain tasks, such as terrain recognition and surface monitoring. Since the data acquisition system under discussion is primarily intended for e-bikes, having the ability to collect data about the motor and the battery is crucial. The DAM was extended with a dedicated data reader component that retrieves data from the control unit of the motor through BLE application programming interfaces (APIs), granted after agreements with the manufacturer. Some of the data recorded include the battery charge level, total distance traveled, residual range, battery capacity, current level of assistance, speed, rider power, motor assistance, cadence, and error code in case of failures of the motor and the battery. The sample rate of these variables varies from 1 Hz to 3 Hz depending on the type of variable. A comprehensive summary of the dataset schema is described in Table 1.

The developed system is housed in a 3D-printed case with a size of $94 \times 83 \times 39\,\mathrm{mm}^3$, designed to protect the electronics while allowing the device to be securely mounted on the bike handlebars. A picture of the DAM and the OM is depicted in Figure 4.

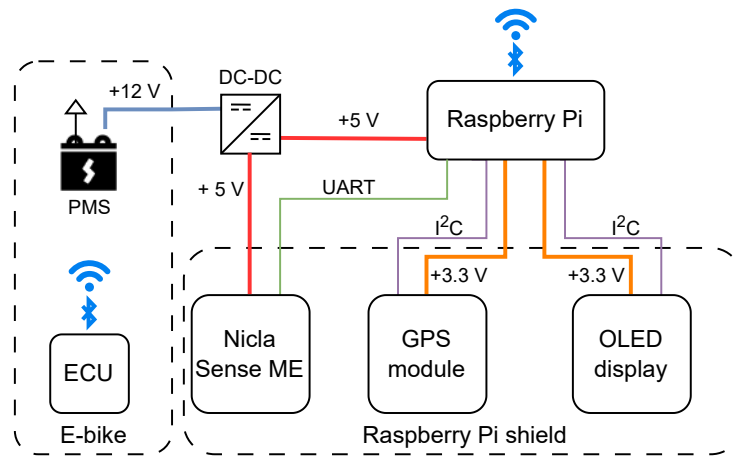A block diagram of the power distribution of each module is shown in Figure 5.

The power consumption of the developed system, as measured by providing the DC power supply from a Keysight N6705C power analyzer, is 3.5 W on average over a 30 min period, with short, periodic peaks of 5 W when the camera is operated.

Table 2 compares the proposed system to others found in the literature [7–9], highlighting the most prominent features of every system.

Table 1. Dataset schema, reporting the name, the source, and the format of the data.

Variable	Data Source	Python Data Type
Datetime	Raspberry Pi	float
Timestamp	Nicla Sense Me	integer
X-axis acceleration	Nicla Sense Me	integer
Y-axis acceleration	Nicla Sense Me	integer
Z-axis acceleration	Nicla Sense Me	integer
X-axis angular rate	Nicla Sense Me	integer
Y-axis angular rate	Nicla Sense Me	integer
Z-axis angular rate	Nicla Sense Me	integer
Temperature	Nicla Sense Me	integer
Relative humidity	Nicla Sense Me	integer
Atmospheric pressure	Nicla Sense Me	integer
Pressure-estimated altitude	Nicla Sense Me	integer
Air quality index	Nicla Sense Me	integer
MOx raw resistance	Nicla Sense Me	integer
GPS time reference	GPS Module	float
GPS status	GPS Module	str
Latitude	GPS Module	float
Longitude	GPS Module	float
GPS-estimated speed	GPS Module	float
GPS-estimated altitude	GPS Module	float
State of charge	E-bike ECU	integer
Total distance	E-bike ECU	float
Residual range	E-bike ECU	integer
Battery capacity	E-bike ECU	float
Level of assistance	E-bike ECU	integer
Speed	E-bike ECU	integer
Rider power	E-bike ECU	integer
Motor assistance	E-bike ECU	integer
Cadence	E-bike ECU	integer
Errors	E-bike ECU	integer
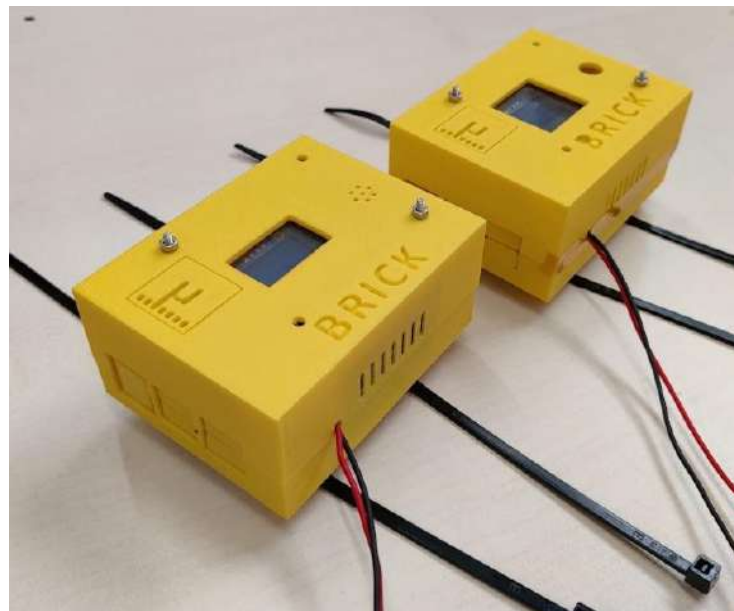
Figure 4. Picture of the developed system housed in the custom 3D-printed case. The two versions show different designs of the holes to ensure the operation of the BME688 sensor. The plastic ties are used to secure the demonstrator on the bike handlebars.

Figure 5. Block diagram of the OM with some details on power supply distribution and communication protocols.

Table 2. Comparison of different state-of-the-art systems found in the literature with the developed demonstrator.

	[7]	[8]	[9]	This Work
On-board sensors	-	PM, distance, IMU	PM, temperature, humidity, barometer, GPS	IMU, magnetometer, temperature, humidity, barometer, MOx, GPS, camera
Processing Unit	Andriod Phone, IOIO board	Android Phone, STM32 Cortex-M3, FPGA	UDOO Neo Full	Raspberry Pi 3B+
Power Consumption	-	-	-	<5 W
Max sampling rate	1 Hz	0.03 Hz	0.025 Hz	100 Hz
Storage	Remote	Remote	Remote, local	Local
Integration with e-bike ECU	Yes	No	No	Yes
Information recorded	Location, altitude, level of assistance	Location, PM	Location, PM, temperature, humidity, atmospheric pressure, formaldehyde	Location, altitude, level of assistance, state of charge, 3-axis acceleration, 3-axis angular rate, temperature, humidity, atmospheric pressure, air quality index, rider power, cadence, battery capacity, residual range, speed, motor assistance, motor and battery errors, total distance
Smartphone required	Yes	Yes	No	No
Wireless connectivity	3G	4G, BLE	GMRS, WiFi	WiFi, BLE, LoRa (to be implemented)
Size	-	-	$190 \times 150 \times 70\,mm^3$	$94 \times 83 \times 39\,mm^3$

4. Field Results

The system was installed on e-bikes during the Giro-E, a cycling event which took place on the days and roads of the Giro d'Italia in May 2024. This event enabled field testing and real-world data collection during a campaign that lasted 22 days, with different environmental conditions, several users with different backgrounds in cycling, from novices to professionals, and a set of tracks with different levels of difficulty.

In what follows, data acquired in different stages of the event will be presented. At each stage of the competition, the e-bikes were produced by the same manufacturer, run by the same team, and the demonstrator was located under the seat of the bike, in a position as consistent as possible. The team included a professional cyclist (PC), the same person for every track, whose duty was to coordinate the overall team, and several users with different levels of cycling experience, scaling with the complexity of the track. The bicycles were all of the gravel type, varying in size depending on the user riding them. All the data reported in the following are averaged on a 30 s time window to suppress spurious readings and filter very fast dynamics, if not otherwise stated.

One of the most interesting tracks to analyze is the one in Livigno (Sondrio, Italy). It is a notoriously difficult stage and involves semi-professional cyclists (SPC). In Figure 6, the altitude is reported against the time of the day along with the state of charge of the e-bike battery, which varies with the color of the scatter plot of the altitude. Figure 6a depicts the state of charge and altitude of the PC, while the same information is showcased for the selected SPC in Figure 6b. The altitude reported, as well as the overall outline, is consistent for the two demonstrators. The height difference is around 2300 m, while the length of the track was calculated from GPS data to be 69 km. These findings are in line with the roadbook provided by the organizers. During the race, the PC used less support with respect to the SPC, as can be seen by the residual state of charge, which is higher in the left plot. Furthermore, it appears that the SPC started using support from the battery first.


(a)   **(b)**
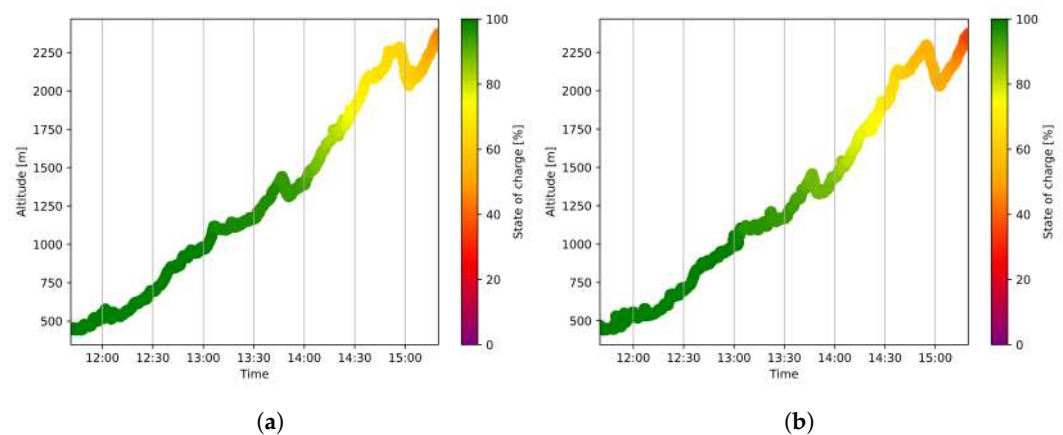
Figure 6. Battery capacity and altitude as a function of time at the Livigno track. (**a**) Professional cyclist data. (**b**) Semi-professional cyclist data.

The usage of the battery and the overall state of the support from the motor can be checked by looking at the level of assistance, another parameter provided by the bike motor system to the demonstrator, reported in the same fashion as before in the plots of Figure 7. The level of assistance is measured in 6 steps, where 0 means no assistance from the motor and numbers from 1 to 5 indicate increasing levels of assistance. The PC plot in Figure 7a shows that the average level of assistance on the first half of the track is significantly lower for the PC with respect to the same period of time in Figure 7b, which is the SPC one. Comparing the two plots section-by-section, the very sharp and long incline starting around 13:30 stands out. The PC likely suggested that users make use of their motors for the follow-up, as the incline is steeper on the second half of the track. Furthermore, the SPC used the same level of assistance for a long time over the whole track, while the PC, probably not needing as much assistance as the SPC, alternated motor support from the

motor mostly to test the bike out. This is particularly evident in the last section, where the PC sprinted to the fullest down the decline around 15:00 and then alternated support on the hardest section of the track following the last decline. The data obtained over time and reported in the plots in Figure 7 are aggregated in plots in Figure 8 to better show the motor usage of the two cyclists.

(**a**)   (**b**)
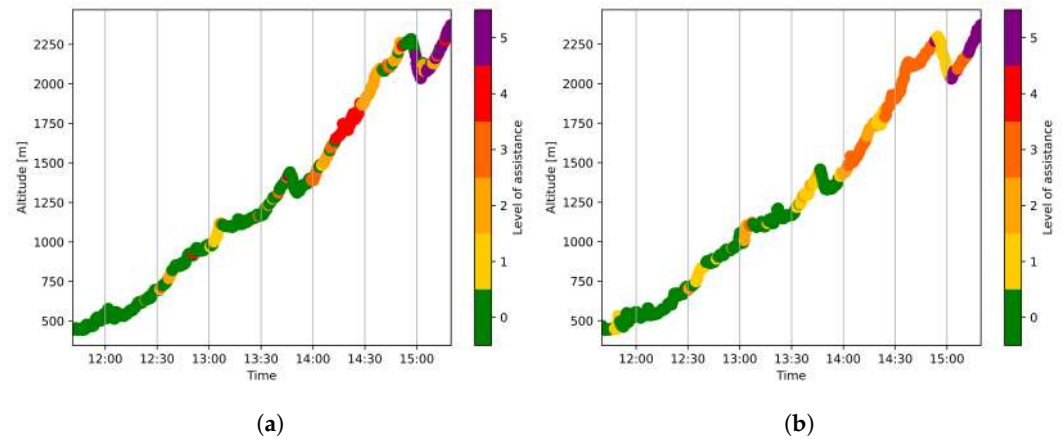
Figure 7. Level of assistance and altitude as a function of time at the Livigno track. (**a**) Professional cyclist data. (**b**) Semi-professional cyclist data.


(**a**)   (**b**)
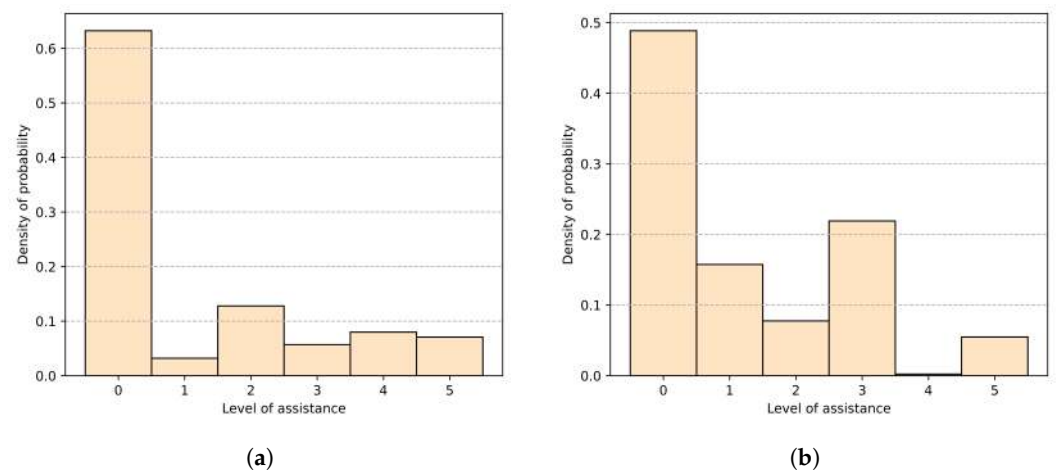
Figure 8. Distribution of the level of assistance over the course of the Livigno track. The probability density must be interpreted as the one to obtain a subtenant integral equal to 1. (**a**) Professional cyclist data. (**b**) Semi-professional cyclist data.

Figure 8a shows that, for most of the track, the PC did not use motor support, and they alternated the level of assistance in a uniform way. As noted above, the PC frequently alternated between motor support levels throughout most of the course. The histogram also shows that the motor support level most used was level 2. Conversely, in Figure 8b, one can see higher usage of the motor assistance by the SPC overall and, specifically, a non-uniform alternating of the levels of assistance. In particular, level 4 was almost never used during the competition, while level 3 was the most used motor support level.

Of particular interest is the estimated power delivered by the rider to the bike reported in Figure 9, another parameter given by the motor interface. By correlating these plots to those of the level of assistance, it is easy to see that a low level of assistance implies higher power delivered by the rider, as might be expected. Comparing the PC in Figure 9a and the SPC in Figure 9b, it is clear that the PC is more consistent in delivering higher power for a longer time than the SPC (for example, in the incline section that goes from 13:30 to 13:45). In general, the power delivered by the PC is higher than that of the SPC, as expected from their professional level. The peak power delivered by the PC is 320 W, while that of the

SPC is 260 W. The distributions of estimated power delivered by the rider are shown in Figure 10. The tails of the distribution in the histogram of the PC displayed in Figure 10a are longer than those of the SPC, as shown in Figure 10b. This can be explained by the fact that the PC alternated the level of assistance frequently during the track, and therefore their power delivery varied more compared to the one of the SPC. This clearly shows the peak power delivered by the PC is higher than that delivered by the SPC, as stated above. The same is true for the average delivered power.

(a)　　　　　　　　　　　　　　　　　　(b)
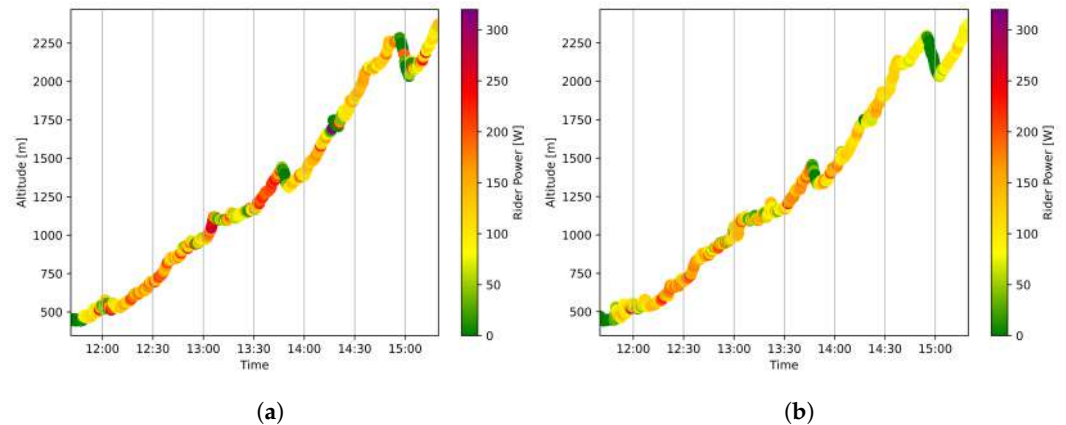
Figure 9. Estimated rider power and altitude as a function of time at the Livigno track. (**a**) Professional cyclist data. (**b**) Semi-professional cyclist data.
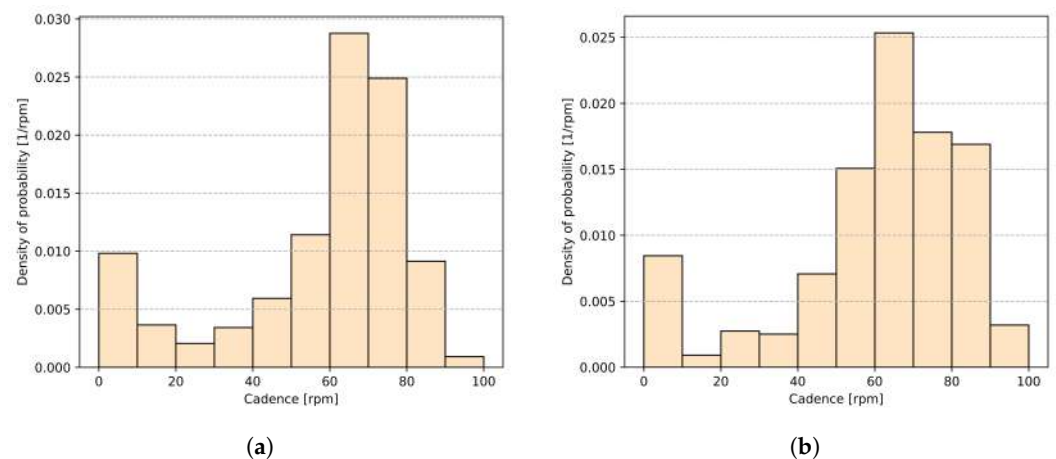

(a)　　　　　　　　　　　　　　　　　　(b)
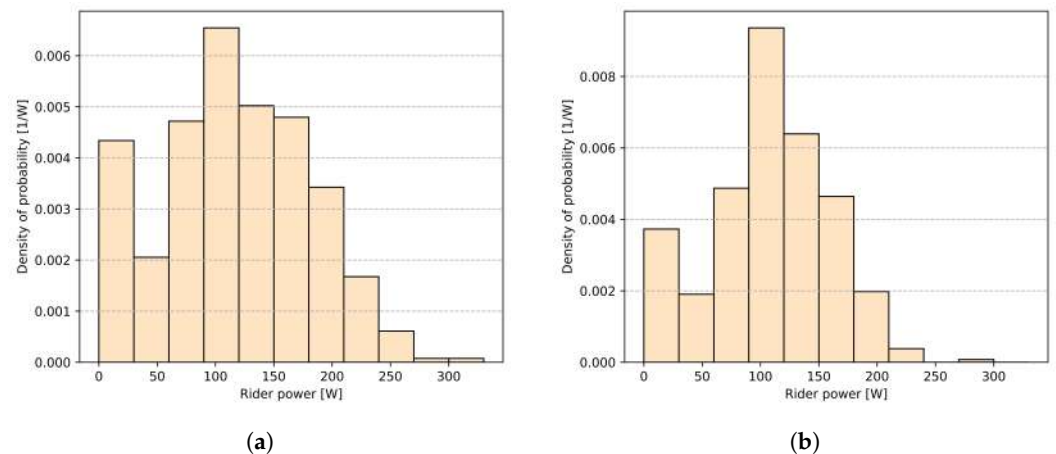
Figure 10. Distribution of the estimated rider power over the course of the Livigno track. The probability density should be interpreted such that the integral under the curve equals 1. (**a**) Professional cyclist data. (**b**) Semi-professional cyclist data.

Another notable piece of information gathered from the motor is the cycling cadence, meaning the number of revolutions of the pedal per minute. Figure 11 displays the aggregated information of the cadence over the track in the form of a histogram for both the PC, in Figure 11a, and the selected SPC, in Figure 11b. The lower bin of the histogram is higher than the ones near it, which can be attributed to declines, where riders typically avoid pushing on the pedals to recover stamina. It is interesting to look at the spread around the average cadence: the PC has a lower spread with respect to the SPC. This is justified since professionals are specifically trained to keep their cadence as constant as possible over the whole ride. In contrast, the SPC's cycling rhythm is more varied.

Another interesting stage to be analyzed is Argenta (Ferrara, Italy). The track is suitable for novice cyclists and is 60 km long with an altitude difference of 100 m. This ride is not of particular interest in terms of dynamics, being mostly flat, but it is located in an urban setting, making it noteworthy in terms of environmental parameters. Specifically, the index of air quality (IAQ) was analyzed for the track. This parameter is extracted directly

from the library provided by Bosch Sensortec to interact with the BME688 sensor and gives a rough estimate of the quality of the air. For the purpose of the current study, it is enough to say that an IAQ above 100 is an indicator of a mildly polluted environment, while a lower value indicates that the air is more breathable. The data employed in this analysis were not averaged using a 30 s time window, but were left raw to have correspondence between the IAQ reading and the GPS reading. In Figure 12a, the map of the Argenta track is reported, with the starting point denoted as A and the finishing point as B. Points A and B are located in urban settings with a higher population density, industrial plants, and more residential areas compared to the rest of the track. The plot in Figure 12b shows the IAQ obtained by the sensor integrated into the demonstrator. The IAQ peaks appear around the beginning and ending sections of the ride, which is precisely where the urban areas of Argenta and Cento are located. In both sections, the IAQ is higher than 100, while it is lower throughout the rest of the track. This could mean that the IAQ is sensitive to the different air quality of the city center and the countryside. However, further studies are needed to confirm this finding, as many parameters vary along with the IAQ. For example, environmental parameters such as temperature and humidity can influence the measurement of this sensor, as shown in many studies [17]. Finally, the sensor employed is designed to be used in an indoor environment, while the application of interest in this work is an outdoor one. For these reasons, no more than an educated guess can be made regarding the association between the IAQ and the environmental pollution on different sections of the track.


(**a**)   (**b**)

Figure 11. Distribution of the cycling cadence over the course of the Livigno track. The probability density should be interpreted such that the integral under the curve equals 1. (**a**) Professional cyclist data. (**b**) Semi-professional cyclist data.

The last dataset that was analyzed was collected on dates other than those of the competition. Specifically, the data were collected in a hill-mountain environment over a two-hour period. The e-bike used for this campaign was different from those employed in the Giro-E, as it was a mountain bike from the same manufacturer. The device was relocated over the center of the handlebars with properly designed, 3D-printed hooks. The main focus of the data collection was to check the ability of the demonstrator to reconstruct different types of roads and their condition based on readings from the inertial sensors integrated into the shield. To reconstruct the road types encountered during the ride, the spectrum of the accelerations and rotations experienced by the IMU was reconstructed over time. The data employed in this stage of the analysis were not averaged using a 30 s time window but were filtered using a third-order Butterworth filter with a cut-off frequency at 40 Hz to get rid of unwanted signals out of the range interesting for the analysis of the road. Figure 13 showcases the spectrogram obtained for the Z-axis accelerations, with the Z-axis lying on the same axis of gravity. The span of time reported in the spectrogram covers different road types and conditions, which are reconstructed by the frames acquired by the

camera attached to the demonstrator. From around 41 min to 60 min, the cyclist was going uphill on a paved road (type A). Between 60 min and 75 min, the road type changed to off-road with irregular pavement, stones, and dry dirt (type B). In the last section, from 75 min to 90 min, the road type was again flat pavement, but the rider was going downhill (type C). The sections with a magnitude around zero, in dark purple, indicate the rest sections, where the e-bike was still (type D). By looking at the spectrogram and not counting the D sections, one can clearly see the three different sections related to the different types of roads, mostly identified by a higher or lower magnitude, with a peak around 10 Hz. B sections have a higher magnitude, which is expected because the irregular terrain would have the biggest impact on the accelerometer. The spreads around the peaks over time are also higher in these sections compared to the others. The second highest peaks are obtained in C sections. This can be attributed to the fact that, going downhill, the system oscillations are composed of higher harmonics and the overall speed is higher, resulting in a large spread and a significant amount of peaks over time around the same frequencies as before, although still lower in intensity compared to B sections. Excluding D sections, A sections have the lowest magnitude. When the rider is moving uphill, the overall speed of the bike is lower and oscillations are usually induced by the handlebars rotating around the Z-axis, meaning that the accelerations generated over the same axis have lower frequency components and overall lower magnitudes, if any. Overall, the system seems to be able to differentiate between different pavement types, road conditions, and user behaviors. Further analysis might be of interest to specialise an algorithm to differentiate between road types.
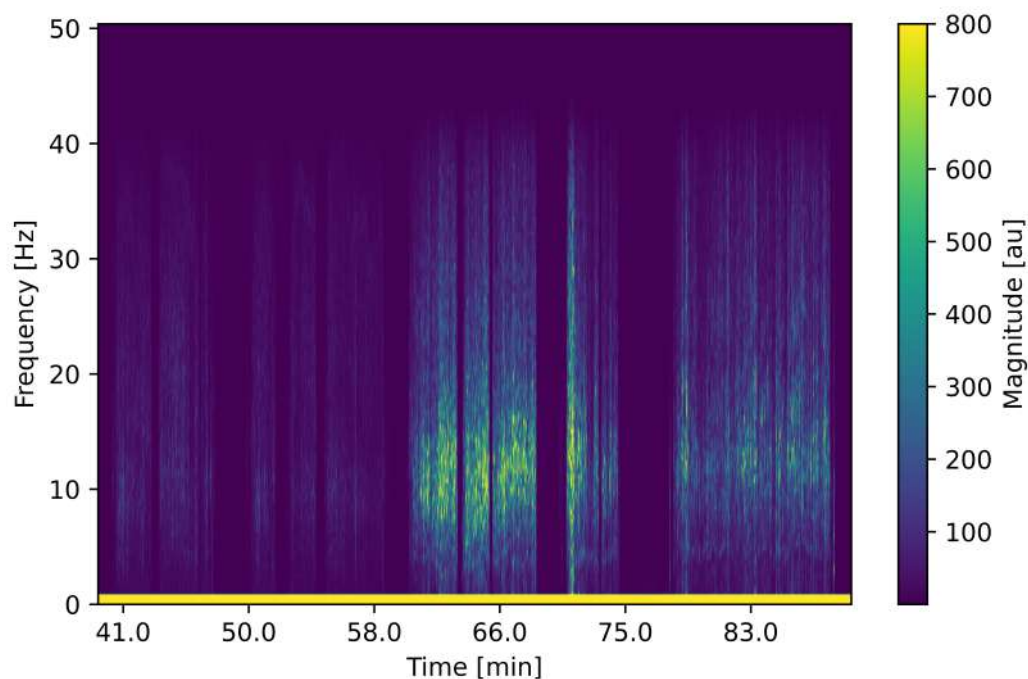

(**a**)
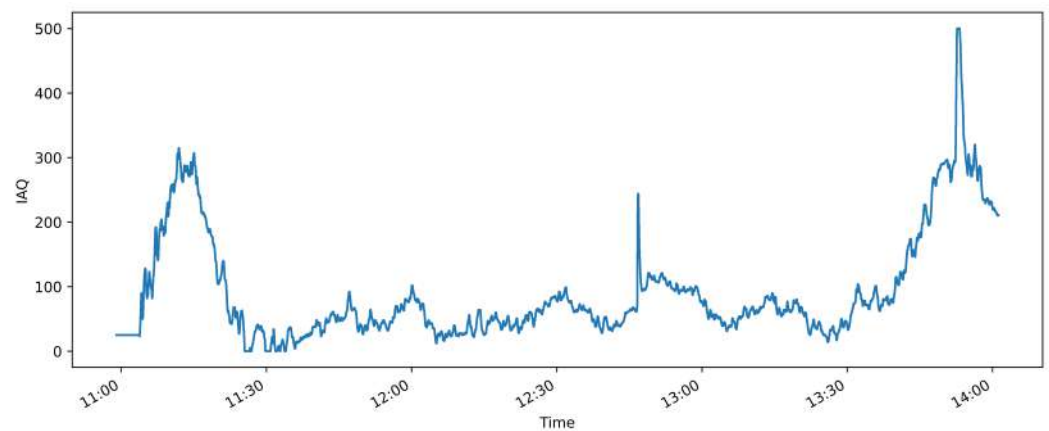
(**b**)

Figure 12. Satellite and environmental data collected at the Argenta track. (**a**) Reconstructed satellite map of the Argenta track based on integrated GPS readings. (**b**) Measured index of air quality plotted over the Argenta track.

Figure 13. Spectrogram of a section of a generic ride with different road compositions.

5. Conclusions

The growing market of e-bikes is paving the way towards a new era of sustainable transportation. To optimize e-bike performance, enhance rider safety, and gain a deeper understanding of user interactions, the availability of high-quality data is crucial. This paper presented a novel, multi-module data acquisition system specifically tailored for e-bikes, designed to collect a wide range of parameters related to vehicle dynamics, environmental conditions, and rider behavior.

The proposed system leverages a centralized architecture, where a high-performance, single-board computer serves as the central data acquisition module. Such a unit gathers data from several sensors, by exploiting different communication interfaces. Data from accelerometers, gyroscopes, speed sensors, and batteries may be exploited to track the e-bike kinematics and open up the possibility to assess the quality of the pavement and to generate warnings in case of fall of the rider. On the other hand, sensors such as temperature, humidity, barometers and air quality, may provide additional insight on the environmental conditions. The data acquisition system is implemented by means of a robust Python-based application, leveraging the versatility and broad library ecosystem provided by such a programming language. By carefully considering factors such as expandability and reusability, the software architecture is designed to accommodate future enhancements and modifications.

A preliminary test campaign has been performed on the developed demonstrator. In particular, some samples of the system have been tested, with professional and semi-professional riders, during the so-called Giro-E, a non-competitive race which took place on the same road as the Giro d'Italia, in May 2024. In conclusion, the multi-module, data acquisition system presented in this work offers a powerful tool for advancing the field of e-bike technology. By providing comprehensive, high-quality data from heterogeneous sensors, this system can significantly contribute to the development of more efficient, sustainable, and enjoyable e-bike experiences.

Author Contributions: Conceptualization and methodology, L.G. (Luigi Gaioni) and M.V.; software, P.L. and M.V.; formal analysis, L.G. (Luigi Gaioni) and M.V.; investigation and validation, L.G. (Luigi Gaioni), A.G., L.G. (Luca Ghislotti), P.L. and M.V.; data curation, P.L. and M.V.; writing—original draft preparation, L.G. (Luigi Gaioni), P.L. and M.V.; all the authors contributed to writing—review and

Abbreviations

AI	Artificial intelligence
APIs	Application programming interfaces
BLE	Bluetooth Low Energy
BSEC	Bosch Sensortec Environmental Cluster
CBSE	Component-based software engineering
CI/CD	Continuous integration and continuous delivery
COBS	Consistent overhead byte stuffing
CPU	Central processing unit
CRC-8	Eight-bit cyclic redundancy check
CS	Charge station
DAQ	Data acquisition
DAM	Data acquisition module
ECU	Electronic control unit
FIFO	First-in-first-out
GIL	Global interpreter lock
GNSS	Global Navigation Satellite System
GNU	GNU's not Unix
GPS	Global Positioning System
GPIO	General purpose input/output
IAQ	Index of air quality
ICT	Information and communication technology
IMU	Inertial measurement unit
IoT	Internet of Things
IPC	Inter-process communication
I^2C	Inter-integrated circuit
LoRaWAN	Long-range wide-area network
LPWAN	Low-power wide-area network
MCU	Micro controller unit
MM	Motor module
MOx	Metal oxide
MQTT	Message queuing telemetry transport
MOST	Centro Nazionale per la Mobilità Sostenibile (National Center for Sustainable Mobility)
OM	On-vehicle module
OLED	Organic light-emitting diode
OS	Operating system
PC	Professional cyclist
PM	Particulate matter
PMS	Power management system
PNRR	Piano Nazionale di Ripresa e Resilienza (National Recovery and Resilience Plan)
PPG	Photoplethysmography
SBC	Single-board computer
SD	Secure digital
SDLC	Software development life cycle
SEMS	Smart e-bike monitoring system
SPI	Serial peripheral interface

SpO$_2$	Blood oxygen saturation
TM	Tire module
UART	Universal asynchronous receiver/transmitter
UML	Unified Modeling Language
USB	Universal serial bus
V2X	Vehicle-to-everything
VOC	Volatile organic compound
WM	Wearable module
WP	Work package

References

1. Koszowski, C.; Gerike, R.; Hubrich, S.; Götschi, T.; Pohle, M.; Wittwer, R. Active mobility: Bringing together transport planning, urban planning, and public health. In *Towards User-Centric Transport in Europe: Challenges, Solutions and Collaborations*; Müller, B., Meyer, G., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 149–171. [CrossRef]
2. Papadakis, D.M.; Savvides, A.; Michael, A.; Michopoulos, A. Advancing sustainable urban mobility: Insights from best practices and case studies. *Fuel Commun.* **2024**, *20*, 100125. [CrossRef]
3. Elvik, R. The non-linearity of risk and the promotion of environmentally sustainable transport. *Accid. Anal. Prev.* **2009**, *41*, 849–855. [CrossRef]
4. Martinez, L.M.; Pritchard, J.P.; Crist, P. Shared mobility's role in sustainable mobility: Past, present, and future. *Annu. Rev. Environ. Resour.* **2024**, *49*, 191–222. [CrossRef]
5. Torre-Bastida, A.I.; Del Ser, J.; Laña, I.; Ilardia, M.; Bilbao, M.N.; Campos-Cordobés, S. Big data for transportation and mobility: Recent advances, trends and challenges. *IET Intell. Transp. Syst.* **2018**, *12*, 742–755. [CrossRef]
6. Delfini, L.; Spahiu, B.; Vizzari, G. Visual analytics for sustainable mobility: Usability evaluation and knowledge acquisition for mobility-as-a-service (MaaS) data exploration. *Digital* **2024**, *4*, 821–845. [CrossRef]
7. Kiefer, C.; Behrendt, F. Smart e-bike monitoring system: Real-time open source and open hardware GPS assistance and sensor data for electrically-assisted bicycles. *IET Intell. Transp. Syst.* **2016**, *10*, 79–88. [CrossRef]
8. Lai, P.C.; Huang, H.Z.; Sheu, M.H.; Wu, C.M.; Le, J.T.; Chen, T.H. Bike sensor system design for safety and healthy riding. In Proceedings of the 2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Taichung, Taiwan, 19–21 May 2018; pp. 1–2. [CrossRef]
9. Aguiari, D.; Delnevo, G.; Monti, L.; Ghini, V.; Mirri, S.; Salomoni, P.; Pau, G.; Im, M.; Tse, R.; Ekpanyapong, M.; et al. Canarin II: Designing a smart e-bike eco-system. In Proceedings of the 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 12–15 January 2018; pp. 1–6. [CrossRef]
10. Verzeroli, M.; Galliani, A.; Ghislotti, L.; Gaioni, L.; Lazzaroni, P.; Re, V. Empowering smart mobility with a component-based data acquisition system for multi-sensor readout. In Proceedings of the 2024 19th Conference on Ph.D Research in Microelectronics and Electronics (PRIME), Larnaca, Cyprus, 9–12 June 2024; pp. 1–4. [CrossRef]
11. Pérez-Zuriaga, A.M.; Llopis-Castelló, D.; Just-Martínez, V.; Fonseca-Cabrera, A.S.; Alonso-Troyano, C.; García, A. Implementation of a low-cost data acquisition system on an E-scooter for micromobility research. *Sensors* **2022**, *22*, 8215. [CrossRef] [PubMed]
12. Tiwari, U.K.; Kumar, S. *Component-Based Software Engineering: Methods and Metrics*; Chapman and Hall/CRC: New York, NY, USA, 2020. [CrossRef]
13. Tanenbaum, A.S.; Bos, H. *Modern Operating Systems*, 4th ed.; Prentice Hall: Boston, MA, USA, 2015.
14. Arduino Nicla Sense ME Product Reference Manual. 2020. Available online: https://docs.arduino.cc/resources/datasheets/ABX00050-datasheet.pdf (accessed on 25 October 2024).
15. Bosch Sensortec BHI260AP Datasheet Self-Learning AI Smart Sensor with Integrated IMU. 2021. Available online: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bhi260ap-ds000.pdf (accessed on 25 October 2024).
16. Bosch Sensortec BMP390 Datasheet Digital Pressure Sensor. 2021. Available online: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmp390-ds002.pdf (accessed on 25 October 2024).
17. Botticini, S.; Comini, E.; Dello Iacono, S.; Flammini, A.; Gaioni, L.; Galliani, A.; Ghislotti, L.; Lazzaroni, P.; Re, V.; Sisinni, E.; et al. Index air quality monitoring for light and active mobility. *Sensors* **2024**, *24*, 3170. [CrossRef]
18. SAM-M8Q Easy-to-Use u-Blox M8 GNSS Antenna Module Hardware Integration Manual. 2020. Available online: https://content.u-blox.com/sites/default/files/documents/SAM-M8Q_DataSheet_UBX-16012619.pdf (accessed on 25 October 2024).