POLITECNICO DI TORINO

Master's degree in Mechatronic Engineering

Master's Thesis

"ITS Technologies in the sustainable mobility scenario. A FCHEV case study"

Politecnico di Torino

Supevisors
Fabrizio Dabbene
Barbara Masini
Giuseppe Napoli

Candidate
Giovanni Vincenzo Mammoliti

Acknowledgements

I want to thank all the people involved in this thesis, all members of CNR IEIIT that have been part of this work, prof. Dabbene, prof. Masini, prof. Mirabella and all the members of CNR ITAE, Ing. Napoli and Ing. Micari, thank you for all your support, availability, expertise and all the things that you have added to me. It has been a long and difficult trip and I also have to thank my family, the supporting column of my life, a unique mixture of different colors without which it would have been impossible to paint this great adventure, all my friends that make me laugh also in the tough moments and that have been close to me even miles away and surely thank to my girlfriend Eugenia, that has always been here to support me, to guide me, following our path together.

Abstract

The purpose of the present work is to analyze, through a case study, the possibilities offered by ITS systems in new sustainable scenarios related to zero-emission mobility. This thesis has been carried out in collaboration with CNR IEIIT and CNR ITAE. The work is divided in four sections: "ITS systems", "Case study", "Development of the communication platform" and "Road Test and future scenarios". The first part explores the relevance of ITS technologies, that are advanced applications that aims to provide innovative services related to different modes of transport and traffic management, allowing safer, smarter and more efficient use of transport networks, and their use in different fields of application from logistics to automotive. After the introduction, an example of the application of ITS in the city of Turin is presented which shows the correlation of these systems with the issues related to the mitigation of climate change and the achievement of the goal of safer and more efficient transport systems. Turin is among the most active cities in Italy for the search for sustainable and intelligent solutions for mobility, so it was considered useful to represent this scenario. Moreover, in 2011, Turin began the process of defining the strategy to become a smart city, whose main elements are public transport monitoring, traffic monitoring and control, travellers information services. This process is managed by a public company, called 5T which deals with ITS in order to improve individual and public mobility taking into account the importance of smart roads and the MaaS theme.

The point of intersection between ITS and integrated services connected to the mobility, as reported in this first chapter, is represented by MaaS, that is the future concept of mobility in which private and public transportation services are integrated in one service, where the most important role is the final user. Maas constitutes, therefore, the possibility for the user to use in a simple and integrated way all the mobility related services (public transport, taxi, bike sharing, ride sharing and so on) through a single digital app that suggests the best solution for the needs of the user offering payments, planning and booking and in order to choose always the best option we have to be aware of what is the traffic situation in that moment and the key concept is: communication. MaaS potentiality is strictly related to the role of public governance, pursuing some fundamental objectives like accessibility, inclusion, innovation and promotion of a more sustainable mobility.

Sustainable mobility means also green mobility, adopting low or zero emission vehicles, that are presented in this first part as BEV, FCEV and FCHEV, showing the main differences between them and introducing the hydrogen production-related technologies and potentialities for mobility. The policy maker represented by the Public Governance pursue social and sustainability goals, encouraging new mobility services in a sharing and green scenario, also through the definition of rules, for these reasons at the end of section 1 it is presented the normative context with the main ITS/green mobility related directives as 2010/40/EU and DAFI.

In chapter 2 there is the presentation of our case study, that is the development of a communication technology platform for a Fuel Cell Hybrid Electric Vehicle developed by CNR ITAE of Messina. The concept behind this vehicle is to overcome the problems related with strictly electric or hydrogen powered vehicle giving a zero emission "hybrid" solution. In this section there is an overview about the main features of the vehicle (battery pack, fuel cell system, power train), including the operating principle of the fuel cell and energy management strategies related to the state of charge of the batteries. The green production of the hydrogen is crucial to achieve the climate objectives that the world is indicating, in this context the I-Next (InnovatioN for green Energy and eXchange in Transportation) project takes place in the Capo d'Orlando experimental site that is presented at the end of section 2.

Considering the ITS context depicted in Chapter 1- the new possibilities offered by communication technologies applied to automotive industry and the importance of such strategies to achieve essential social, economic and climatic goals- and the description of the vehicle in Chapter 2- a fuel cell hybrid vehicle that makes use of "green hydrogen" directly produced on site - the aim of this third section is to show the potentialities offered by the interaction between the vehicle, different devices and a system able to interact with all these subsystems exploiting different type of communication technology. The adopted microcontroller (the system that takes care of interfacing with other devices to provide some output to the final user) is an STM32F429ZI, a very versatile and highly configurable system. In this section there is a presentation of the adopted communication protocols, SPI, I2C, CAN and the computational results with the used devices (RTC, OLED, SD Card, BMP180, GPS). The last part of this thesis is focused on the test on the vehicle, analyzing the CAN communication, through which we have access to a number of messages, from the State of Charge of the vehicle to the pressure inside the hydrogen tank or other important messages; all these parameters will be helpful in the communication environment from a vehicle to another vehicle or to an infrastructure or to a pedestrian exploiting these data in real-time to take crucial decisions in future scenarios that are the last topic of this chapter.

Contents

List of Figures

List of Tables

Acronyms

AVM	Automatic vehicle monitoring
BOP	Balance of Plant
CAN	Controller Area Network
CIM	Centrale Integrata della Mobilità
CNG	Compressed natural gas
CPU	Central Processing Unit
CS	Chip Select
C-V2X	Cellular Vehicle-to-Everything
FCS	Fuel Cell System
GLOSA	Green Light Optimized Speed Advisor
GPS	Global Positioning System
GUCE	Gazzetta Ufficiale dell'Unione Europea
IRQ	Interrupt Request
ITS	Intelligent Transportation system
LTE	Long Term Evolution
MaaS	Mobility as a Service
MEA	Membrane Electrode Assemblies
MISO	Master Input Slave Output
MOSI	Master Output Slave Input
OLED	Organic Light Emitting Diode
PLN	Piattaforma Logistica Nazionale
PUMS	Piano Urbano della Mobilità Sostenibile
RAM	Random Access Memory
RTC	Real Time Clock
SCL	Serial Clock
SDA	Serial Data
SMA	Smart Mobility Agency
SMR	Steam Methane Reforming
SOC	State of Charge
SS	Slave Select
TAPE	Turin Action Plan for Energy
URR	Ultimately Recoverable Resources
VAL	Véhicule Automatique Léger
VMS	Variable Message Sign
V2I	Vehicle-to-Infrastructure
V2P	Vehicle-to-Pedestrian
V2V	Vehicle-to-Vehicle

Chapter 1: ITS systems

1.1 Overview about the Intelligent Transportation System

An intelligent transport system is an advanced application that aims to provide innovative services related to the different modes of transport and traffic management, allowing safer, smarter and more efficient use of transport networks. ITS is the integration of technologies in the field of telecommunications, electronics, information technology with transport engineering, for the planning, design, operation, maintenance and management of transport systems. This integration is aimed at improving driving safety and the safety of people, the safety and protection of vehicles and goods, optimizing the use of natural resources and respecting the environment.
The opportunity to share a huge amount of information between vehicles, road users or with road infrastructures is the best way to improve traffic efficiency in terms of time, costs, safety and sustainability. This thesis work is focused on the opportunity given by "zero-emission fuel vehicles" to communicate with the surrounding world in a scenario in which the data that the vehicle is able to send to other systems are merged with the information coming from the outside to the vehicle, for example from a charge infrastructure.
The interaction between information technology, telecommunications and multimedia makes it possible to deal with the problems of public and private mobility in an innovative way, developing solutions based on safety, efficiency, economy and respect for the environment in an organic and functional way.
The fundamental elements (as shown in Figure 1) that make up an ITS system are:
1. "Data collection: First level systems, or the systems placed on the territory in order to collect the data and information necessary for the CIM to operate;
2. Processing and management of the information contained in the collected data: Data processing systems (data fusion), or the IT layer that extracts the summary information and provides decision support data;
3. Information distribution: Supervision and info-mobility system, that is the information layer of man-machine interface, which on the one hand provides the views of summary and detailed information and, on the other, allows operators to enter other information (typically relating to accidents, construction sites, etc.) and to manage user information systems (news and automatic information). " [1]

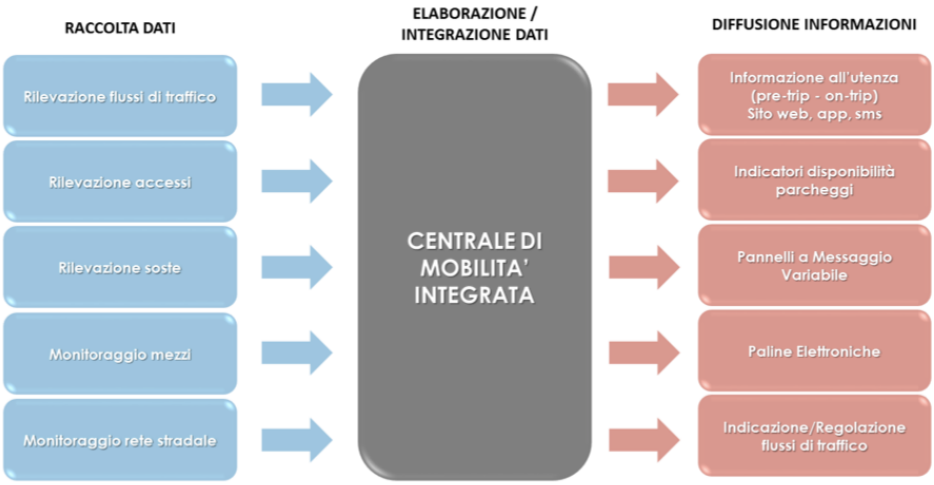

Figure 1: Integrated Mobility Central for data collecting/dispatching

The main components of ITS are two: the first one is the traffic management centers (an example is shown in Figure 2), the nerve centers of the systems, this is where command and control takes place, the second important feature is the communication, made of electronics, software and network gear that runs that communication, that's how we bring all the devices back to the traffic management center.

Figure 2: Traffic Management Center

ITS are, therefore, a technology that allows traffic monitoring and mobility management (even the public one), which deals with citizen information, it can be used for advanced vehicle control to improve safety and also for control of heavy transport fleets and emergency management. With the use of ITS we have obtained: reduction of travel times by 20%, reduction of congestion by 15%, reduction of polluting emissions by 10 %. [2]

ITS require very low investments that are not comparable with those required for large infrastructural works that often characterize the world of transport; they are also limited investments with a very quick return on investment. Italy is an active country in this sector as well as a leader in Europe. The improvement of the attractiveness and quality of the local public transport service also passes through the ITS, obtaining an improvement in the travel times of the lines, the regularity of the service; the companies that operate the local public transport service can implement with these systems a more efficient management of both their fleets and local public transport but also of the internal processes and of the personnel assigned to provide this service; citizens can certainly perceive an improvement in the comfort and quality of the public transport service and finally an effect of a general nature: the adoption of these systems can lead to a better balance in what is defined as a modal choice, that is, it can favor the choice to travel with local public transport between origin and destination or in any case to make transfers using different services. Some examples of services that can be implemented using ITS technologies are monitoring intended as the adoption of systems such as AVM, AVL control and management of the public transport fleet, for example the detection of the actual use of the service: being able to know how many people there are on board at all times is a fundamental fact in order to better plan the service and also to allow public transport companies to implement savings where there

is little demand for the service, for mobility. “AVM is a system that allows you to monitor different quantities relating to moving vehicles (for example position, distance, speed, component diagnostics). It is a system similar to the telemetry used in motor racing and is used in most cases for the management of vehicles in local public transport fleets”.[3]
Despite the continuing economic crisis that our country has gone through in the last decade, a survey presented by TTS Italia in 2016 entitled "Il Mercato dei sistemi intelligenti di trasporto in Italia, Quadro attuale e prospettive" showed that the sector of ITS is constantly growing, as presented by Figure 3, both in terms of companies that have entered this sector, both in terms of dedicated personnel and above all in terms of turnover which for 2020 is about 1.7 billion euros [4]

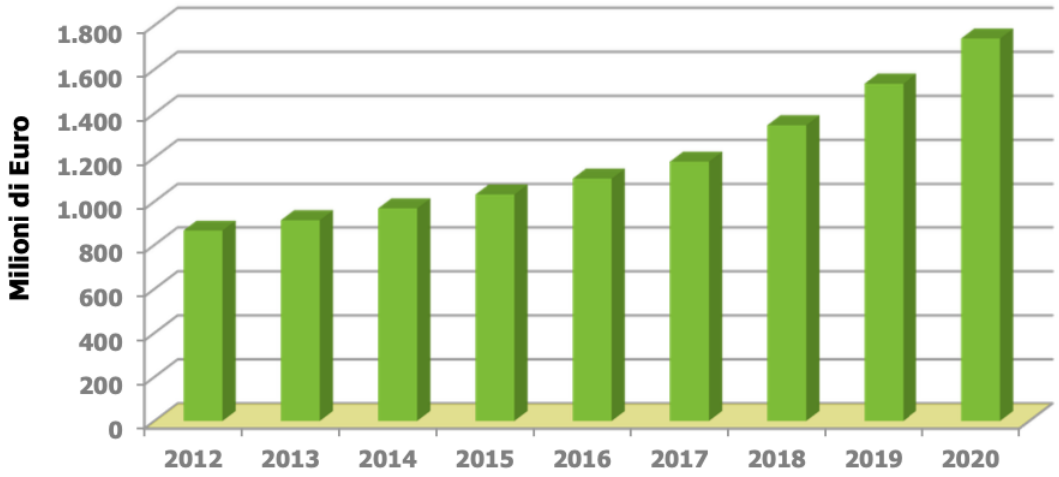

Figure 3: Growing market of ITS system

In Europe, Italy is one of the countries with the highest traffic density and as regards safety, the data tell us that the number of road accidents is still very high and that the social cost for them is about 1.1% of National GDP, about 19.3 billion euros. Thanks to the use of ITS on the Italian territory, substantial improvements have been obtained in terms of traffic and travel times (for example for the city of Turin the first factor increased by 17% while the second decreased by the same percentage, bringing also to a 10% decrease in climate-altering gas emissions).
The goal in relation to Covid was to allow the movement of citizens and goods in a safe manner, trying to identify a new point of balance between the demand for mobility and the supply of transport. In this context, new technologies for the management, information and planning of journeys play a fundamental role in order to make mobility more efficient and smarter. A survey was conducted by TTS Italia to be able to group together proposals and new ideas for ITS technologies related to the pandemic emergency. The entire survey was developed nationally and internationally. Most (85% at the Italian level and 93% at the international level) replied that smart working will continue to be the norm in the workplace over the next 2 years. The main effect of smart working is the decrease in commuting from home to work with the remodeling of peak moments. Among the questions asked was also asked how the role and way of using collective transport (bus, metro, train, plane, taxi, etc.) could change and 95% of the Italians interviewed replied that the population will tend not to use collective transport if not forced and that being the alternative to public transport the private car is expected to increase traffic. The last fact that I want to mention, which I find particularly interesting, is that eighty-five percent of the sample questioned indicated that it is necessary to rethink the model of urban mobility from an intermodal perspective through the massive use of ITS and MaaS services. ITS represents the next generation of transportation where it’s not just about getting from point A to point B but getting there in a safer, greener and more convenient way. By connecting and distributing intelligent processors inside vehicles and throughout the transportation infrastructure we can

transform transportation in profound and unimaginable way. Intelligent travel will be multimodal enhancing all forms of transportation whether by car, rail or air and travelers will be able to remain seamlessly connected across every means of travel. Future vehicles will be equipped with a processor performing many tasks simultaneously such as improving navigation, communication all while enabling safer and more efficient travel. Our vehicles will be able to connect over a variety of wireless standards to the internet to intelligent traffic system and even to other vehicles. As society develops the roads becomes more crowded and cars grow in number, there are numerous variables and unforeseeable situations hiding on the unlimited number of roads. It's very important to have an intelligent and smart solution to help us solving all these problems and also to make people's life safer, faster and easier. The technologies used in the ITS field are the data collection technologies (infrastructure-based technology, vehicle-based technology), communication technologies (wireless communication) and database management. Interconnections no longer just of telephones or computers, also affecting mobility. Vehicles, passengers or pedestrians become part of a complex network that exchange information constituting the so-called smart city. These smart cities should solve various problems, vehicular traffic and its correct management and the protection of those who use means of transport or pedestrians. Therefore, through this network, efficiency and safety problems can be solved. Connected intelligence will extend beyond vehicles, existing and emerging wireless standards will enable vehicles and devices across the transportation infrastructure to communicate with each other enabling automated intelligent real time decisions to optimize travel, for example strategically placed safety cameras will identify congestions or accidents, automatically alerting emergency personnel as well as rerouting traffic by notifying approaching vehicles and updating nearby digital roadside signs. Intelligent traffic lights will become more prevalent and smarter than today detecting real-time flow from each direction and automatically changing signals to improve traffic flow. Distributed intelligence that is secure and reliable is the key transforming the entire traffic infrastructure from being passive and unaware to a proactive intelligent connected sense and response system that improves travel in every way. All the intelligent transportation technology will make travelers to be better informed and make safer, more coordinated and smarter use of transportation networks while saving money, fuel and time. In the network society things and people are connected, it feels natural that cars and vehicles in general fit into this picture as there are many interesting applications that can be enabled by connecting them. We can distinguish among different types of connectivity depending whether people, vehicles or the network infrastructure are being connected, for example with V2I (vehicle to infrastructure) we can enable a number of applications such as fleet management, infotainment, remote car diagnostics management. With V2I we can even enable cars to communicate with the network and cloud applications within the network infrastructure. Intelligent Transport Systems will help to reduce the wasted time and energy by optimizing trips, reducing congestion, improving vehicle and driver performance and fostering better management of the transportation system as a whole. The optimization of the transport system will result in energy savings, lower pollution levels and reduced environmental impact [5]. The smart city aims to make mobility interconnected, safe, efficient, comfortable and above all respectful of the environment. Smart city also means multimodal mobility, this means a greater variety of choices to go from point A to point B, therefore also means the development and expansion of the local public transport offer, innovation of existing infrastructures (for example new electric charging stations). In order to balance the so-called "modal split" between public and private mobility and in order to reduce pollution, it is necessary to encourage innovative public mobility, with zero impact, whether it is electric or hybrid. We can identify three sustainability objectives to be achieved: the economic one, in order to make the most of resources; the social one, ensuring accessibility even to the weakest and environmental subjects. "Annually, WHO estimates that millions of deaths are caused by the effects of air pollution, mainly from noncommunicable diseases. Clean air should be a fundamental human right and a necessary condition for healthy and productive societies. However, despite some improvements in air quality over the past three decades, millions of people continue to die prematurely, often affecting the most vulnerable and marginalized populations," [6]. On 25 September 2015, a list of 17 objectives

making up the Sustainable Development Agenda was adopted by the United Nations General Assembly. Target number 11, which puts cities at the center, aims to reduce emissions of substances harmful to the environment and to provide sustainable, safe, efficient and affordable transport. Each of these goals is closely linked to the others, creating better cities from renewable energy, developing innovative infrastructures to fight climate change together.

1.2 Relevance of ITS in different field of application

In this chapter I would like to show the great opportunities offered by ITS and what are the practical improvements that they can offer in different sectors. One of the possible fields in which ITS can boost the innovation is the logistic world. Logistics has always been a sector with a low rate of innovation, and it is estimated that it has a weight on the national economy of about 35 billion euros (2% of GDP), it is, therefore, necessary to modernize the sector also through the use of technology offered by ITS.
The PLN, whose main elements can be seen in Figure 4, is owned by the "Ministry of Sustainable Infrastructures and Mobility and was developed by UIRNet S.p.A. with the aim of increasing the safety and efficiency of national logistics. It has been conceived to concentrate and deliver system services to all operators, becoming the platform for interconnection and direction of data and related processes, it allows continuous and intelligent interaction between those who manage the road and logistic infrastructures, those who transport, load / unloads and checks goods" [7].

Figure 4: PNL

"These, in summary, the services currently provided by the PLN:
• Port Community System (PCS) (development): system that allows public bodies and private operators, who regularly operate in the port, to exchange information in a secure way through the adoption of open protocols, accessible to all interested parties, while ensuring the confidentiality of strategic information that is important to the operators themselves;
• Customs Controlled Corridor (CCD) (in operation): telematic system for the geo-referenced control (with geofencing technology) of goods transport on pre-established routes, ensuring the integrity of transport, both for goods transported by road and for those use intermodal transport. In this way it is possible to relocate and simplify customs operations, in order to free port areas and speed up import operations and, therefore, decrease the overall transit times relating to the supply chains;
• Freight Village System (development): system for the operational management of logistic processes for interports and logistic plates. The development of personal data services and the management of permits is expected in line with the knowledge developed on the PCS;
• Dangerous Goods Base (partially implemented in Messina): service aimed at the operational control of the transport of dangerous goods. The service is aimed at the complete management of the planning, implementation and control phases of road transport of dangerous goods;

• Intermodal Appointment (development): solution that allows the truck driver to book an arrival slot in the port to be served by the terminal and, at the same time, provides the terminal with visibility of arriving vehicles, with automatic control of the completeness of the documentation;
• Security event management (implemented): system that allows the redirection of traffic in the case of critical events (up to the closure) of the logistic node. It is an advanced service that the PLN can provide, even if it provides for the availability of equipped parking areas, where the flows can be redirected. The areas have yet to be built and will have to be built after an analysis on traffic flows and demand. " [7]
One of the possible applications of ITS to logistics is the development of platooning, which consists of a series of vehicles automatically arranged on the same line in which the first vehicle is called "head of the platoon" and the rest of the vehicles are called "followers".

“The potential advantages of platooning are:

- Highway Capacity: vehicles driving in a platoon are able to keep short spacing thus improving the lane occupancy;
- Fuel economy: vehicles driving in a platoon are subject to a reduction in aerodynamic force. The shorter is the distance, the lower is the drag force;
- Safety: truck platooning helps to improve safety. Braking is automatic and immediate; the trucks following the lead vehicle only need one-fifth of the time a human would need to react;
- User comfort: when one vehicle has information about other vehicles on the road, it can decide in advance which action to take. So, it is possible to driver safer and smoother”.[8]

Figure 5: ITS Platooning application

Ericcson is cooperating with Scania in a project exploring the possibilities enabled by connecting trucks to the networks. There are interesting ITS applications also related to improved safety. Active safety can greatly improve the figures for driving-related accidents. By integrating the information from the sensors in the car with communication between vehicles, we can improve the probability of timely detecting a dangerous situation and act accordingly by avoiding the accidents or at least minimizing the consequences. Communication between the vehicles is fundamental because sensors by themselves are not able to detect all risky situations and this type of communication is called V2V, vehicle to vehicle. Now, think if my car could exchange its speed, position, direction and other important information with other vehicles in my surrounding. By doing this, we could become aware

of each other and detect these dangerous situations in advance. Another very interesting area is V2P, communication between vehicles and pedestrians. Many vulnerable road users, such as pedestrians or cyclists already today use an LTE smartphone. Now think if we can connect these users to the cars. For example, a car driver could be aware that in a certain position near to him there is actually a man with his pram that is going to cross the street and if we could make this connection in advance, we could avoid a dangerous situation. “The development of ITS systems has led to a decrease of 51% in deaths, 27% in accidents with fatalities and 19% among all types of accidents” [9].

The number of vehicles on the road are increasing faster than the number of people. From 2016 to 2017 we saw an increase of 1.6% in the number of registered vehicles, compared to a 0.8% increase in the overall population, this is leading to more congested and dangerous roadways. To answer this transportation problem, we shouldn’t continue to do the exact same thing we have always done, which is building more roads. There is not enough space to continue to do that and also, it’s extremely costly and it takes year to see an impact of congestion for new construction. One way we could improve upon and optimize traffic management is by viewing the roadway as a highway for information, versus a highway that carries vehicles. Sidebar radars, toll meters weather stations, traffic cameras, harnessing these data from the roadway will help us optimize traffic both now and into the future. The good news is that the data we need does exist, the not so great news is that it generally resides in very different places and is generally disconnected, leaving us with a fragmented view, at best of what’s happening on the roadways. There are many different transportation data sources that exist, but, again, they live in different places. Without a clear understanding of what’s happening on the roadways, our communities pay for mobility and sometimes with their lives. But what if we could bring all these data together and even let the data be the driver for traffic management? We could enhance situational awareness, leveraging current and historical data (situational data, weather data, traffic data) to respond to the traffic incidents in near real-time, the moment they happen. We could coordinate response across law enforcement, emergency response to respond to an incident on where it is from community feedback. We could assess roadway needs using data points like traffic volume and frequency, weather events, community feedback to understand where we need to do repairs in a predictive way, on the roads. With all this data brought together, you can harness the data to answer some key questions, like how has traffic changed over time? What alternate routes can be utilized? When is congestion expected to peak? Where are the most traffic roadways? Once the data is organized and available for both historic and real-time analysis, we obtain data-driven decisions, helping to make roads and drivers safer.

Many of the discussed applications can already be enabled today with existing LTE technology, however, for some of the most challenging use cases, we need some enhancement in the LTE Radio Access in order to fulfill them efficiently. ITS applications can be very demanding in terms of network load, consider hundreds or even thousands of devices within a limited geographical area, continuously exchanging information with each other, so the question is: how can the network support such high capacity? One way to smartly integrate the cellular wide are LTE communications between the devices, this can be done by taking into account the capabilities of each device, the type of service and also the instantaneous network load situation. Another important type of enhancement is related to smart and efficient radio resource management, there we can combine distributed algorithms implemented in the devices with a centralized optimization performed by the base station.

Transport is the most important and critical sector within cities in the field of green mobility innovation, first of all because it is the first cause of emissions of three types of pollutants, NO, CO, NMVOC and for the emission of fine particles (PM10 and PM 2.5). It is therefore necessary to reduce emissions by creating eco-compatible transport systems, thus encouraging electric mobility or other sustainable mobility solutions, such as hydrogen, promoting innovative technologies and activating incentives and rewarding mechanisms (purchase of vehicles with low environmental impact, passes for local public transport). Noise pollution represents a significant nuisance for citizens due to the high noise emissions along the streets of the city.

Road traffic is the most dominant source of environmental noise in Europe. It is estimated that 125 million people are affected by noise levels from road traffic greater than 55 decibels (dB) Lden, including more than 37 million exposed to noise levels above 65 dB Lden [10].
Estimates show that Turin, for example, is characterized by a very high rate of noise pollution, causing the limit values to be exceeded in many sensitive areas, such as schools, hospitals and nursing homes. In Turin, as can be seen in Figure 6, about 400.000 people are exposed to Lden values higher than 65db .

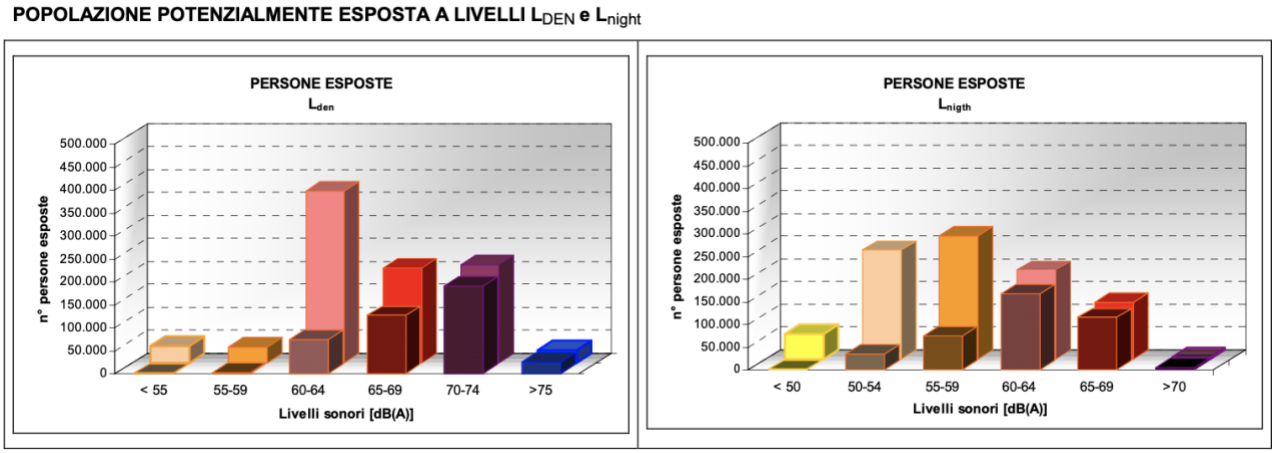

Figure 6: Lden levels for Turin

"In the intentions, the plan for sustainable urban transport should cover the entire urban area and seek to reduce the negative impact of transport, addressing the increasing levels of traffic and congestion, and should be linked with the strategies and plans regional and national. The plan should cover all modes of transport and seek to change the balance in favor of more efficient modes of transport, such as public transport, cycling and walking. One of the fundamental objectives is the creation of a more efficient transport system from an environmental point of view and placed at the service of all citizens, who have a fundamental role to play in their daily decisions, such as the choice of the mode of transport. An essential element of the plan would be the link with the territorial structure ". [11]

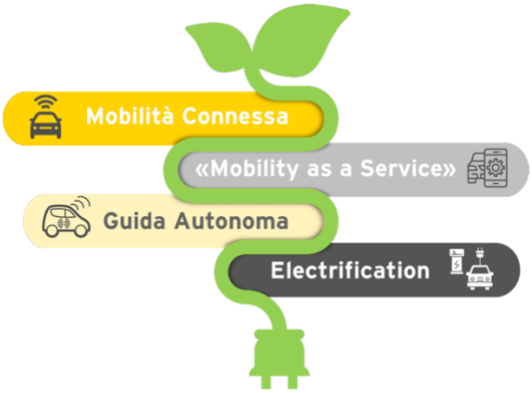

Figure 7: Elements for green and smart mobility

1.3 Example of ITS applications in the city of Turin

Turin is one the most active city in Italy in the research for sustainable and intelligent solutions for mobility, being this city one of the most polluted in Europe.
Italy has lagged behind in the measures to be taken to reduce air pollution and there are 35 provincial capitals that do not respect the limits regarding fine dust, with the negative record that belongs to the city of Turin. It is estimated that over fifty thousand deaths can be traced back to exposure to climate-altering factors, such as fine dust (in particular PM 2.5), nitrogen oxides and carbon monoxide, in our country alone. , globally there are over 7 million a year, effectively making the problem of pollution a silent massacre. A change must be implemented, starting precisely with the strengthening of local public transport and shared mobility, areas in which ITS make their contribution to make the experience of mobility more efficient, reliable and safe for citizens.

Figure 8: Number of cars per thousand inhabitants in different countries

Turin is characterized by approximately 902.000 inhabitants with an area of 190 square kilometers, 18.4 square kilometers of green area, 160.000 trees, 200 km of cycle paths (which with the biciplan want to become 300 kilometers), an estimated pedestrian area of 9.52 square meters per inhabitant, 1.32 square kilometers of restricted traffic area with an estimate of the city's vehicle fleet which indicates the following data: 597.000 cars, 52.482 commercial vehicles, 72706 motorcycles. The estimated average time dedicated to mobility is about 57 minutes a day, with an average length of the journey of 10.1 kilometers, the main reasons for travel are for work or study (35.2%), family management (34, 6%) and free time (30.2%). It is therefore estimated about 2 million trips a day to and from the city with the following modal split: 20.8% on foot, 5.2% by bike, 1.5% by motorcycle, 56.2% by car and finally 16,3% public transport. Line 1 of the underground extends for about 13.2 kilometers and crosses the city from west to south, the line is characterized by light trains without driver and interchanges with surface public transport favoring various means of transport. it consists of a metro line (VAL) characterized by 21 stations covering just over thirteen kilometers, 8 metropolitan train lines, 7 tram lines and 90 bus lines (30% CNG and 4 electric vehicles). It then makes use of various sharing services including 3 bike sharing services (which count about 22600 subscribers), namely Tobike (120 stations and 750 bicycles), Mobike (1500 free-flow bikes) and

Helbitz (500 bicycles for free flow pedal assist). There are 3 car sharing services, Car2GO (400 vehicles), Enjoy (250 vehicles) and BluTorino (400 electric vehicles with 700 charging stations). We must also remember the services of scooters and scooters, such as MiMOTO (about 150 vehicles), ZiGZAG (about 125 vehicles) and as regards the scooters there are 6 active services that can count on about 500 vehicles per operator (Bird, Dott, Helbitz, LiME, BIT Mobility). All the possibilities offered by the city of Turin in the mobility context are summarized in Figure 9. In this context, the concept of mobility detaches itself from that one linked to possession and ownership to achieve shared mobility that can be used as needed. So, the key word is connection and intermodality by developing a connected network of trams, buses and sharing services, all connected by ITS technologies that allow tracking and localization, offering the user a more sustainable mobility also from a safety point of view, overcoming the concept of private mobility linked to the possession of a car and allowing the diffusion of more effective economical systems and therefore accessible to more people.

Figure 9: Green, electric and sharing application in the city of Turin

The Piedmont area, in particular the City of Turin, characterized by intense anthropization and industrialization, is protected by mountain ranges that make the dispersion of pollutants difficult, resulting in worrying concentrations, in particular of PM10, as shown in Figure 10. Over the past 30 years, the air quality in Turin has improved significantly. In fact, since the 1970s, the problems of sulfur dioxide, lead, benzene and carbon monoxide have been solved; just think that in 1973 the daily maximums of sulfur dioxide exceeded the threshold of 2,000 mcg / mc and now we are at values around 50 mcg / mc. However, the objective and significant improvement in air quality is not yet sufficient to comply with the new limits introduced by European legislation to protect human health and the environment and there are still strong critical issues for: Fine Suspended Particulate (PM10), Dioxide Nitrogen (NO2) and Ozone (O3) [12]. As regards PM10 and PM2.5 fine particles, transport represents the main emission factor of these pollutants (about eighty-five percent), as well as nitrogen dioxide due essentially to vehicular traffic, with particular reference to diesel cars.

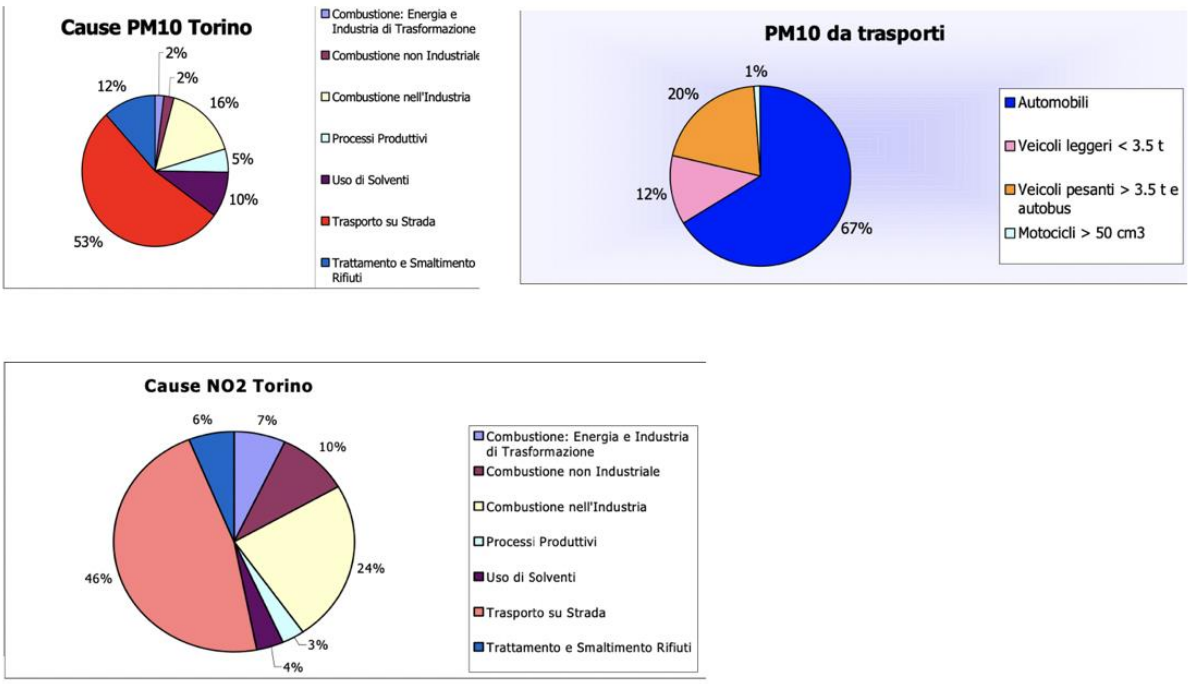

Figure 10: Pollution in the city of Turin

The city of Turin has set objectives to be achieved with regard to the issue of sustainable mobility, that are, to improve the transport of people by reducing the use of private cars and developing intermodality of public and shared services and also freight transport using low or zero emission vehicles, having signed in 2019 the "Freight Quality Partnership" which is an agreement between the city of Turin, the Chamber of Commerce and the main transport, trade and industry associations in order to improve environmental conditions linked to the distribution of goods, replacing polluting vehicles and promoting the use of ITS. Improving the urban distribution of goods is essential to reduce vehicular congestion. In order to improve, therefore, the efficiency of mobility and to offer a better service, Turin makes use of different technologies, one is linked to info-mobility (“Muoversi a Torino” is the only portal), allowing users to always choose the optimal route using the various means of transport, such as bicycle, tram, bus, metro or car. In order to manage all the information related to vehicular traffic, the city has a Traffic Operations Center which has the role of monitoring the various instruments installed in the city, including about 3000 traffic sensors, 26 information panels, 71 urban cameras on 23 intersections, also controlling the access management in the restricted traffic area. The plant analyzes and checks the data, always guaranteeing priority to public transport. Air quality, noise, congestion and safety are the key issues for planning mobility. The PUMS is a set of actions aimed at introducing a type of mobility (public and private) that is more accessible, safe and more respectful of the environment thanks to a modal rebalancing of the different types of transport, favoring soft mobility (for example the bike) , increasing the possibilities of interchange and sharing, making the public transport service more punctual and accessible and also improving private vehicle mobility, increasing safety and reducing the harmful effects on the climate. The main objectives for the city are to guarantee and improve the accessibility of people, improve air quality, encourage the use of collective transport, guarantee efficiency and safety of the road system, govern the mobility through innovative technologies. The PUMS for Turin therefore provides 78 new bike stations, 148 bike to rail, the extension of metro 1 and the birth of metro two (26.2 kilometers, 32 new stops in 2030), 19 kilometers more than tramway network, 7 interchange nodes between metro, railway and ring road system and the increase in electric cars (25% of those in circulation by 2030).

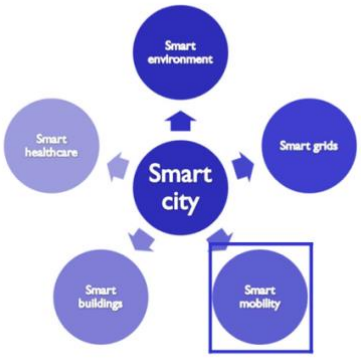

Figure 11: Smart city elements

5T is a public company that deals with ITS systems. 5T deals with ITS in order to improve individual and public mobility by taking into account the issue related to smart roads and the theme of MaaS. The shareholders of 5T are: GTT, the City of Turin, the Province of Turin and the Piedmont Region. The company, founded in 1992, was responsible for the development and management of the traffic operation center of the 2006 Turin Winter Olympic Games. 5T manages the traffic monitoring center in the Turin metropolitan area. The Turin traffic center manages all data related to mobility to improve safety and efficiency, in particular, it uses 1500 inductive loops for real-time detection of traffic flows, 36 electronic gates for the limited traffic area, 20 VMS for information on parking, 18 extra-urban VMS, 36 VMS positioned at the ZTL gates. In 2011, Turin began the process of defining the strategy to become a smart city. In the development of a smart city, whose elements are highlighted in Figure 11, a fundamental role is that of public transport, which must exploit, in order to increase its performance, the new technologies for planning and monitoring the fleet through the AVM system, collecting data and analyzing them and then providing to the end user info-mobility services in real time.

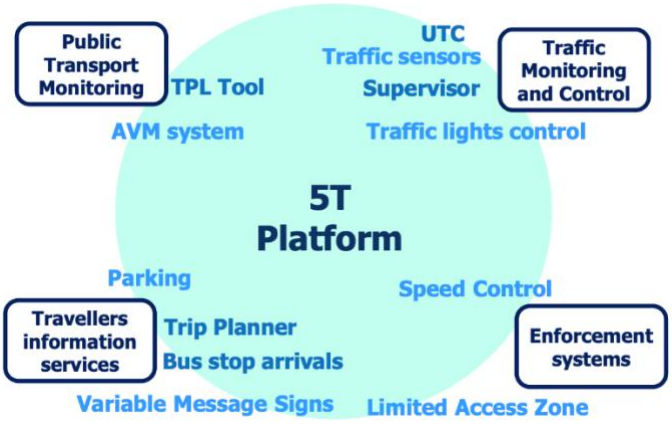

Figure 12: 5T Platform.

Turin has been awarded with the Global Road Achievement Award 2019 in the context of "urban mobility" in Las Vegas. Turin, with the support of TTS Italia, participated with the project "The Turin Smart Road Project: the role of public authorities as key enablers of innovative mobility services based on ITS". Several case studies were presented, starting from ITS to autonomous driving, passing from road safety also linked to vulnerable subjects with the development of the vulnerable road user warning, which informs you of the presence of pedestrians or cyclists or other soft mobility users;

then there are projects such as GLOSA to communicate to vehicles the ideal speed to get to the next green light. There are also projects related to roadside assistance that improve real-time communication between the hospital and the rescue vehicle guaranteeing priority to the rescue vehicle. Smart Road intends to facilitate the development of innovation in the field of mobility and transport by allowing the experimentation of autonomous and connected vehicles authorized by MIT on some of the main arteries and reference structures of the city [13]. The first test was made possible thanks to the support of the “Politecnico di Torino” and Danisi Engineering. SMA collaborated in the development of this project as a facilitator for the implementation of new technologies, such as V2V or V2X. Harvard Alumni Entrepreneurs, an international alumni organization of the American university with a focus on entrepreneurship and innovation, has chosen Turin as an urban laboratory model for the first appointment of 'Smart Cities', a series of online events on smart cities [14]. The Smart Road is a new concept of intelligent road to allow communication and interconnection between the vehicles that travel along it. In smart roads, to facilitate flows and transport, weather and traffic detection systems must be implemented so that travelers can request information on road conditions, traffic or other particular situations in real time. Furthermore, smart roads aim to provide services for diverting traffic flows in the event of accidents, suggestions of alternative trajectories, management of accesses, parking lots and supplies, timely interventions in case of emergencies. Anas intends to invest a total of 1 billion euros in the coming years for digitization projects, including 3000 km of smart roads [15]. To solve the problems related to the emission of pollutants, noise pollution, vehicular traffic and congestion, Turin is developing a series of measures that make the city one of the most active in Italy in the ITS sector. There are, in fact, many projects for the future promoted and developed by the city of Turin, among which we can certainly mention the STEVE project which has as its primary objective the diffusion of light electric vehicles, SOCIALCAR which wants to develop a network for intelligent and connected mobility.

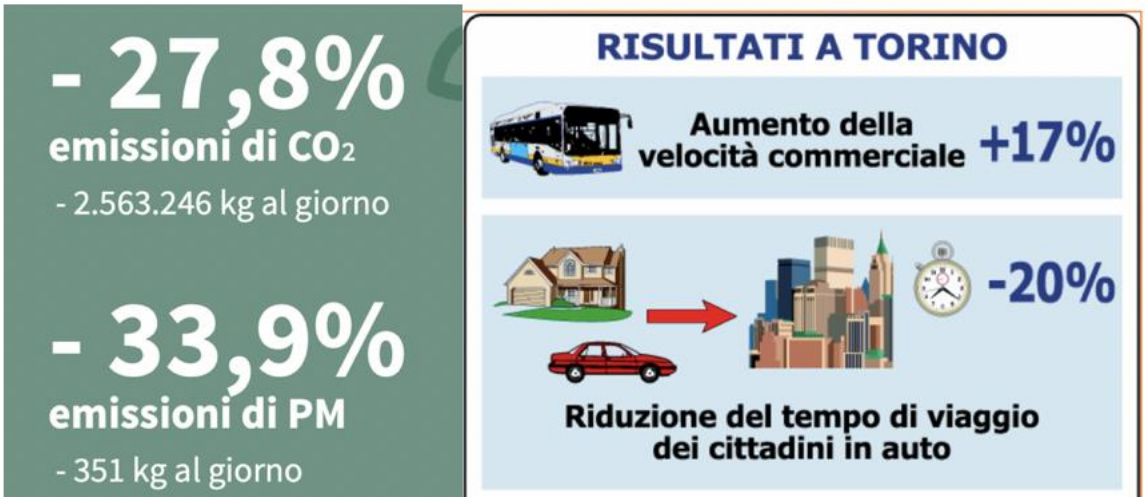

Figure 13: Impact of ITS in Turin

1.4 MaaS and sustainability

The point of intersection between ITS and the integrated services connected to the mobility is represented by MaaS, by taking advantage of the information provided by ITS it's possible to manage through a single portal the mobility demand.
Maas is the future concept of mobility in which private and public transportation services are integrated in one service, where the most important figure is the user. "With the concept of MaaS we refer to the gradual transition to the consumer's preference to use on-demand services to obtain overall cost advantages for the use of mobility solutions, time spent on End-to-End travel and experience overall. In particular, the prospect of buying a vehicle is less attractive today, due to elements such as congestion of the roads and the onerousness of maintaining a vehicle, in terms of cost and time. Last but not least, global mobility is embracing mobility service solutions also to achieve the planet's climate sustainability objectives, reducing emissions and pollution and the number of total vehicles in circulation. The main development segments of this area are: multimodality of transport, understood as a set of smart solutions capable of integrating End-to-End public and private transport (up to the last mile) ensuring complete coverage for the movement of the person, subscription services (on demand mobility), which aim to offer on the market no longer just vehicles, but a set of services, such as maintenance, insurance, within the price of a service that can be stipulated for a short period, medium or long according to the consumer's needs ". [16] MaaS constitutes, therefore, the possibility for the user to use in a simple and integrated way all the mobility related services (public transport, taxi, bike sharing, ride sharing and so on) through a single digital app that suggests the best solution for the needs of the user offering payments, planning and booking and in order to choose always the best option we have to be aware of what is the traffic situation in that moment and the key concept is: communication. MaaS potentiality is strictly related to the role of public governance, pursuing some fundamental objectives like accessibility, inclusion, innovation and promotion of a more sustainable mobility. MaaS solutions allow to follow the mobility needs during all the trip phases influencing individual mobility choices, in this scenario, the digital transformation can speed up the achievement of sustainability goals. The Public Administration, in this context, can play the role of regulator and policy creator, stabilizing rules and deciding the relationship between transport service operators, MaaS operators and users. The definition and the realization of the right policies can let the MaaS to be adopted on a vast scale, with economic, social and climatic impact index of the different mobility choices, encouraging sustainable transport.

Ruolo	Attore	Prodotto/soluzione principale	Tipo di business	Modello di business	Accordi	Esempio della modalità di esercizio	Scopo geografico
MaaS Operator	Pubblico/privato	Offre la soluzione di trasporto che integra almeno tutto il trasporto pubblico. Integra soluzioni di "Digital ticketing"	B2C	Generalmente sottoscrizione, offre diversi modelli di offerta per gli utenti, può integrare l'offerta di trasporto digitale dei singoli operatori di trasporto	Integra diversi operatori di trasporto pubblici e provati tramite il MaaS Integrator	Sottoscrizione mensile	Urbano Bacino Regionale Nazionale
MaaS Integrator	Privato	Offre una soluzione tecnologica per integrazione dei dati dei diversi attori che esercitano il	B2B B2G	Operation & Maintenance: costi di manutenzione, evoluzione, assistenza tecnica,	Integra gli operatori di trasporto pubblici e provati fornendo al MaaS	Soluzioni di integrazione delle informazioni di mobilità, delle informazioni	Urbano Bacino Regionale Nazionale

		trasporto, eventualmente integra soluzione white label per il MaaS Operator		conduzione applicativa.	Operator dati per consentire l'erogazione dei servizi.	su titoli di viaggio e tariffe	
Operatore di trasporto	Pubblico	Esercita materialmente il trasporto, offre servizi digitali al City user per l'informazione e acquisto di titoli di viaggio	Risorse pubbliche e G2G	Vendita del servizio agli utenti e revenue sociali: ottimizzazione, capillarità, pervasività, qualità del trasporto pubblico	MaaS Integrator	Rivendita abbonamenti, informazione sullo stato dei mezzi, gestione del sistema di bigliettazione elettronica	Urbano Extraurbano
Operatore di trasporto	Privato	Esercita materialmente il trasporto, generalmente tramite servizi di sharing	B2C	Vendita del servizio agli utenti, normalmente sulla base del numero di corse o sul tempo di utilizzo del mezzo offerto	MaaS Integrator	Vendita di servizi di carsharing etc.	Urbano Extraurbano
Policy maker	Pubblico	Raccoglie ed analizza i dati nel rispetto della sicurezza e imparzialità. Propone soluzioni per incrementare l'utilizzo e migliorare l'efficienza del trasporto pubblico	G2G	Definisce le Policy di trasporto per incrementare l'utilizzo e migliorare l'efficienza del trasporto pubblico	MaaS Integrator, MaaS Operator, Operatori di trasporto	Analisi della domanda segmentata sulla base di profili, analisi dei fabbisogni degli utenti	Regionale Nazionale

Table 1: Roles in MaaS

The autonomous vehicles will bring a further innovation, in a scenario in which a vehicle is just a component of a set of alternatives. Moreover, they will optimize and simplify the sharing mobility in the urban context. The technology at the base of autonomous vehicles will let vehicles and infrastructures to be integrated generating benefits in terms of climatic sustainability also by using different indexes to let the user to be aware of the emissions related to a particular trip selection.
It's possible, therefore, to see a direct connection between the climatic subject and possibilities offered by ITS.

1.5 Low/Zero emission vehicles

The problems of pollution and therefore the objective of the European Parliament to mitigate the climate change and the need to be independent from fossil fuel led the attention to new possibilities for the vehicle mobility such as electric vehicles. Nowadays, in Italy, electric mobility represents just the 0.2% of the total circulating cars as shown in the figure below.

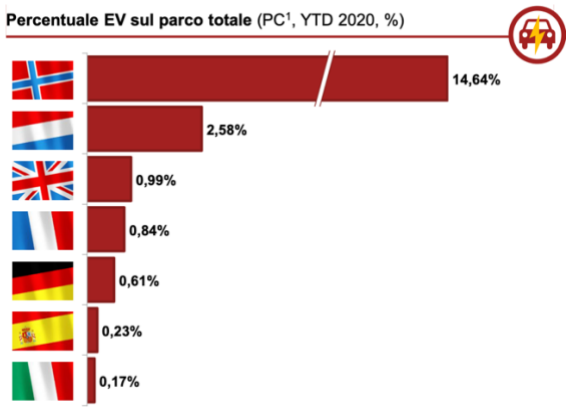

Figure 14: Percentage of electric vehicles in different nations

In Italy we have 8500 charging infrastructures (95% are AC charge point with power between 22 and 43kW), while the High-Power Chargers are still quite rare as shown below. The diffusion of electric cars in Italy is mainly related with regional incentives, the distribution is: 68% North Italy, 24% Central Italy, 8% South Italy. We can distinguish different kind of connector, Type 1 consists in 5 connections, 3 power connections (L1, N, PE) and 2 communication connections (PP CP), this kind of connector is the standard one for North America and Japan and it can be used only for single phase charge, Type 2 has 7 connections overall, 5 for power (L1, L2, L3, N, PE) and 2 for communication (PP, CP), this is the European standard for the alternating current charging station, it can be used both for single phase and three-phase charge and it is the most adopted connector in European electric vehicles, Type 3C has 7 connections, 5 for power and 2 for communication, it is becoming obsolete, it can be used both for single and three-phase charge, Type 3A has 4 connections, 3 for power (L1, N, PE) and 1 for communication (CP), it is dedicated to light electric vehicles, like scooter.

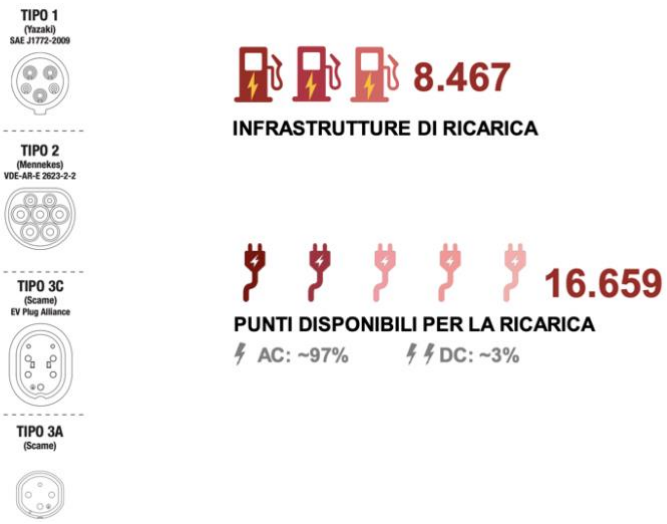

Figure 15: Charge infrastructure

The problem of AC is the charging time, a solution to overcome this problem can be represented by DC fast charge connectors. CHAdeMO is the most common standard for DC fast charge, it is used, for example, by automotive industries as Nissan, Mitsubishi, Peugeot, Citroen; the vehicles equipped with CHAdeMO has 2 connectors; fast DC and AC connector. Another type of DC connector is CCS (Combined Charging System) Combo2 consists in a single connector that allows both DC fast charge and AC charge, it is realized started from connector Type 2, it is adopted by BMW and Volkswagen. One of the most common example of zero emission automotive related technology is the one adopted in BEVs (Battery Electric Vehicles), we identify with these kind of vehicles the full electric vehicles, that are fully operated through power generated by a battery (generally lithium ion), in other words a vehicle that is moved by the power produced by an electric motor fed by a battery. This kind of vehicle doesn't have the typical combustion engine that is replaced by a zero-pollutants emission electric motor. These vehicles, being free from any combustion, are noise-free, reducing the noise pollution. The disadvantages are mainly two: the small autonomy range compared to the distribution and number of charge stations and the other problem is the long time period that this kind of vehicle takes for recharging. The advantages offered by another category of green vehicles, such as FCEV (Fuel Cell Electric Vehicles) are the zero emission power production and the reduction of noise as in the previous case, but FCEV differs from BEV because they are characterized by a larger autonomy and a limited recharge time that can be compared to the one of an internal combustion engine vehicle but here the problems are the very high costs and the substantial absence of infrastructures for the hydrogen refill. In this scenario an efficient approach can be the hybrid one, with the FCHEV (Fuel Cell Hybrid Electric Vehicles), they have a lower recharge time (compared with BEV), greater autonomy, lower cost than a pure hydrogen vehicle.

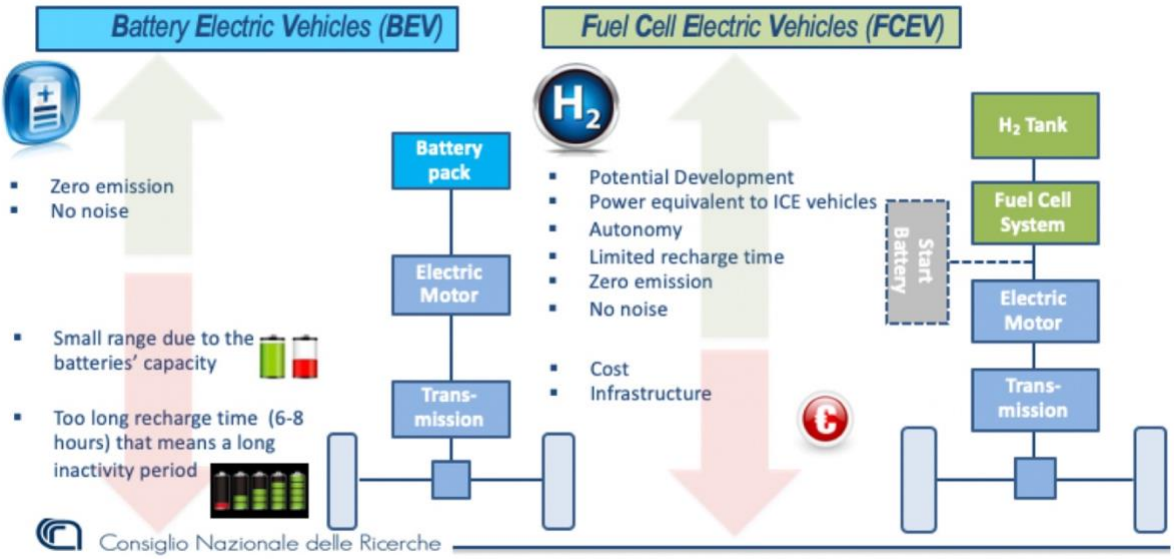

Figure 16: BEV and FCEV

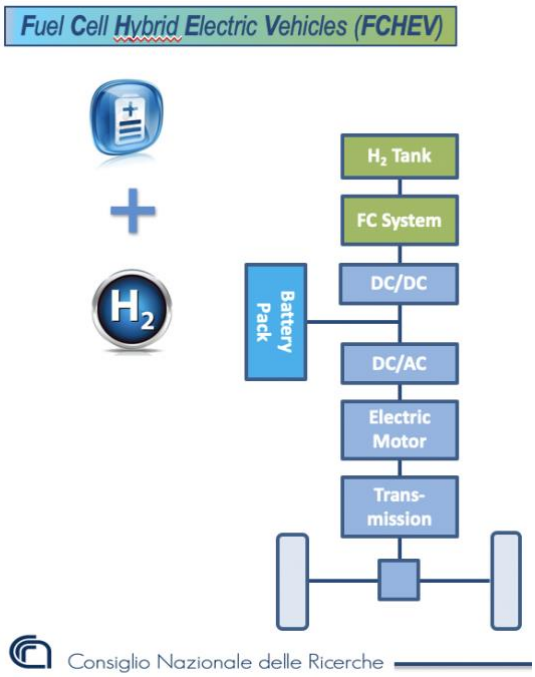

Figure 17: FCHEV

At this point the question is: how the electricity that move my car is produced? When we use something, we normally throw it away. This is the linear economy, and this means that the product has an end of life. Our economy can't go this way forever. We have to make everything capable of being recycled. This is circular economy. Solar energy is key. Nowadays 80% of the energy comes from fossil fuels that are finite resources. Solar energy will be the energy source of the future because it is plentiful. We use a little bit of solar energy for our use, but the ambition is to make fuels starting from the sun. In the next paragraph the hydrogen potentialities for the mobility will be examined in dept.

1.6 Hydrogen Mobility

Hydrogen is one of the “fuels” used to move the FCHEV and it is one of most interesting green fuels for the future in different application of the mobility. Surely the use of hydrogen as a "fuel" for mobility is not suitable for light transport, by which I mean cars, motorcycles, bicycles because in that sector there is already a formidable competitor which is electric mobility. The gas or coal must be brought to the place of production, that is to the power plant, so if I stay in a remote area it is a rather serious problem; with renewable sources the flexibility of electricity is highlighted, because it is possible to produce electricity where I am because I have a physical flow of some nature that affects me, for example the wind, the sun, the movements of the sea, so I have the enormous advantage of being able to produce it by investing in a machine that produces and distributes it. The electricity grid is the most widespread infrastructure in the world, so it is an already existing infrastructure, to be developed and modernized again, but we can count on something that is already there. It is unthinkable that hydrogen can compete with electrified mobility for various reasons: the first is that electricity already has a distribution network present everywhere, another reason is efficiency, if you took 100 units of renewable electricity for the operation of an electric car, in the various passages I would have losses, from 100 starting units produced for example by a photovoltaic panel at the end 77 of these units come to move the wheels of the electric car, if I repeated the same operation with a hydrogen car, suppose we still have a photovoltaic panel from which we produce energy, the electrolyzer that uses electricity to obtain hydrogen starting from water, the hydrogen obtained must undergo compression before being used with the cell to fuel that again produces electricity that moves the wheels of our car, in this case from the initial 100 units 30 arrive, that is less than the observed data for a battery. From here we deduce that the use of hydrogen for light mobility is unthinkable. The reason for this low hydrogen efficiency is given by the presence of two conversions. Another reason not to forget is that in order to obtain an acceptable volume of hydrogen I have to compress it and this phase also has a high energy cost. Today 70 million tons of hydrogen are produced per year. Let's imagine using this hydrogen to power hydrogen cars, with this amount we could run about 500 million Toyota Mirai in the world. If this quantity of hydrogen were obtained entirely by electrolysis, 3.7 PWh of electricity would be needed, which corresponds to 15% of the current global electricity production and over 7 times the production from photovoltaics. These data should make us think, asking us as a question whether it makes sense to produce hydrogen for this purpose or is it better to use electricity in a battery.

When we talk about heavy transport the speech changes radically, hydrogen can be a perspective. Heavy transport means buses, trucks, ships. It can be thought to produce it in large localized production centers (ports, bus parking lots) and to distribute it directly where production takes place without the need for a distribution network. The interesting thing is that heavy vehicles such as buses, ships and trucks typically run defined routes. The prospects of hydrogen as regards heavy transport are very interesting, in order to produce hydrogen in significant quantities we will have to have a huge surplus of renewable electricity that we do not currently have, so if we want to get hydrogen to cover an important role in heavy transport we must enormously increase the production of renewable electricity, in particular the cheaper and easily installed technology is photovoltaics which is perfect for the production of hydrogen because in the daily peaks I can produce hydrogen. If we look over time, since civilization left written traces, about 5000 years, energy consumption for the first millennia was substantially negligible and energy was obtained from muscle strength, wind and fire. At one point we discovered a huge treasure under our feet, named fossil fuels and energy consumption in a very few generations took a huge leap. More than eighty percent of energy still today is obtained from fossil fuels, the question is how long it will last. This cannot last much longer, for reasons of available quantities but also for reasons regarding the environmental situation. Each year, it is estimated, we consume 4.4 billion tons of oil, 7.6 billion tons of coal and 2.5 billion tons of gas. The

main problem with burning this huge amount of fossil fuels is that we throw 33.4 billion tons of carbon dioxide into the atmosphere which, as we all know, alters the climate. The problem with carbon dioxide lies in the fact that it is a very discreet gas, that is, since it is not visible to the eye, we are not worried about it too much, becoming an invisible enemy.
"Another interesting sector for hydrogen is the railway sector, in particular passenger transport: about one third of the railways in Italy is dedicated to diesel trains, contributing to a small portion of national transport emissions. In the next ten years, fuel cell trains may become cost-competitive compared to diesel trains, becoming one of the most promising sectors to start the development of a national hydrogen market. In some European countries (e.g. Germany), hydrogen passenger trains are already fully operational and used regularly by travelers. In the United Kingdom and France, some proposals have been made to completely replace diesel trains with hydrogen ones within the next twenty years for the travel of sections that are difficult to electrify "[17]. Alstom introduced the first hydrogen-powered trains with fuel cell systems, covering a 100-kilometer route for passenger service in Germany. These trains have a range of about 800 kilometers and fuel cells occupy the upper part of the trains, guaranteeing performance in terms of speed and range comparable to those using diesel power train systems.

Figure 18: Hydrogen Train Application

"Fuel cell electric vehicles do not generate local emissions such as NO_X and do not emit any CO_2 from the vehicles. On such a tank-to-wheel basis, only FCEVs and battery- electric vehicles (BEVs) are fully CO_2 emission free, unlike other decarbonization options such as biofuels, compressed or liquified natural gas (CNG/LNG), and hybrids. These technologies can therefore only serve as bridge technologies until BEVs and FCEVs are ready in large numbers, which is not an attractive value proposition for investors. For a fair comparison with diesel and gasoline vehicles, not only tank-to-wheel, but also well-to-tank emissions should be considered – i.e., the emissions from fuel production. Well-to-tank emissions for diesel and gasoline include the emissions from oil extraction, transport, refining and processing, and distribution to the fuel station. For BEVs, well-to-tank emissions depend on the power mix and hence on the country where the vehicle is charged. For FCEVs, well-to-tank emissions depend on the hydrogen production technology. When hydrogen is produced from natural gas with CCS, FCEVs emit 40 to 45% less emissions than vehicles with internal combustion engines (ICEs). As production from hydrogen shifts to full decarbonization, FCEVs will fall in emissions until they are virtually CO_2 free" [18].

	Current role	Demand perspective	Future deployment	
			Opportunities	Challenges
Cars and vans (light-duty vehicles)	11200 vehicles in operation, mostly in California, Europe and Japan	The global car stock is expected to continue to grow, hydrogen could capture a part of this market	Short refueling time, less weight added for energy stored and zero tailpipe emissions. Fuel cell could have a lower material footprint than lithium batteries.	Reductions in fuel cell and storage costs needed
Trucks and buses (heavy-duty vehicles)	2500 forklifts 500 buses 400 trucks	Strong growth segment; long-haul and heavy-duty applications are attractive for hydrogen		
Maritime	Projects for small ships and on-board power supply in larger vessels	Maritime freight activity set to grow by around 45% to 2030.	Hydrogen and ammonia are candidates for both national action on domestic shipping decarbonization and the IMO Greenhouse Gas Reduction Strategy	Reduction in storage cost
Rail	Two hydrogen trains in Germany	Rail is a mainstay transport in many countries	Hydrogen trains can be most competitive in rail freight (regional lines with low network utilization)	Rail is the most electrified transport mode; hydrogen is an option to replace non-electrified operations
Aviation	Small demonstration projects and feasibility studies	Large storage volume and redesign would be needed for pure hydrogen, making power-to-liquid and biofuels more attractive for this mode	Hydrogen: together with batteries, it can supply on-board energy at ports and during taxiing	Power-to-liquid: currently 4 to 6 times more expensive than kerosene, decreasing to 1.5-2 times in the long-term

Table 2: Hydrogen applications in different fields

Over the past 15 years, FCEV buses have been operating across more than 8 million km in Europe, proving that the technology works, is flexible, operational and safe. A total of 77 FCEV buses are in operation. FCEV buses have amply demonstrated that they can be used on normal routes, in regular passenger service, without constraints to adapt to selected or customized routes. [19]

Figure 19: Hydrogen bus map

Hydrogen is a gas just like methane but unlike methane, natural sources of hydrogen are quite rare. The use of hydrogen is by no means new, for example hydrogen is needed to synthesize ammonia, used to make ammonium salts and therefore fertilizers. Today hydrogen is produced mainly using methane (CH4), the so-called "grey hydrogen" (if it is obtained from coal it is called brown),

producing CO2 in the chemical process. It is no longer possible to obtain hydrogen in this way; it is necessary to reduce C02 emissions. So, the starting molecule cannot be CH4 but H20, hence water. To make hydrogen from water, machines called electrolyzers that run on electricity are needed. The European goal is to implement the production of green hydrogen, because it is produced from water and electricity and does not produce C02. Of course, electricity must be renewable, otherwise it would make no sense. To make 1 kg of hydrogen I need 50kWh of electricity, which is the average requirement of an Italian family for a week.; about 5.6 / 6kg of hydrogen goes into a hydrogen car, so I need 300kWh of electricity to develop that amount of hydrogen, with an electric car consuming 4-5 times less energy. Hydrogen is not a source of energy but is a so-called energy carrier, that is, something I can produce to store energy and transform it from one form to another. For example, if I have excesses of renewable energy, I can obtain hydrogen from water by releasing oxygen into the atmosphere. Nowadays, hydrogen is mainly produced from methane, obtaining carbon dioxide which is released into the atmosphere and is clearly a process that has a huge environmental impact. This is a problem because we produce too much carbon dioxide negatively affecting the thermoregulation of the planet. It is therefore necessary to change direction, turning to green hydrogen obtained from renewable electricity using electrolyzers that take electricity from photovoltaic, wind, geothermal, hydroelectric and produce hydrogen from water and therefore there is no production of climate-altering gases. One possibility given by hydrogen is to store electricity, that is, we produce electricity from renewable sources, we produce hydrogen that we store in a fuel cell, which is a device that converts hydrogen into electricity, that is, we split the hydrogen into the proton H + and into the electron that circulates in an external circuit and feeds the device (car for example), the proton combines with oxygen and generates water.

The most efficient way to use hydrogen is inside fuel cells which are devices capable to convert the chemical energy contained in the fuel into electrical, thermal energy, reaction products (water). In fuel cells we have an electrolyte that is able to transport either cations or anions, we have the fuel, for example hydrogen and finally the comburent, that is, oxygen. The redox reaction takes place at the anode, the hydrogen loses electrons which are taken out and therefore usable and then return to the cathode where the circle closes, because the H + ions that are produced at the anode and reacting with electrons and oxygen produce water. A cell has a power and voltage limit, if we want higher powers, we need to put a lot of them together, creating a stack. The single cell is called MEA and then joining with other cells forms a stack, the BOP is the electronic and control part that completes the fuel cell system. There are fuel cells of different types and a possible division can be made based on the working temperature, low, medium and high temperature. Clearly the operating temperature influences the different uses and fields of action of the cells themselves making them more suitable for certain applications, for example those fuel cells that work at a lower temperature need to be able to stop, switch off and even switch on again very quickly, in order to which the fact of having low working temperature also allows a quick start-up. One of the main advantages is the absence of polluting and climate-altering emissions, subsequently having high energy efficiency and low charging times, comparable to the charging times of an internal combustion car.

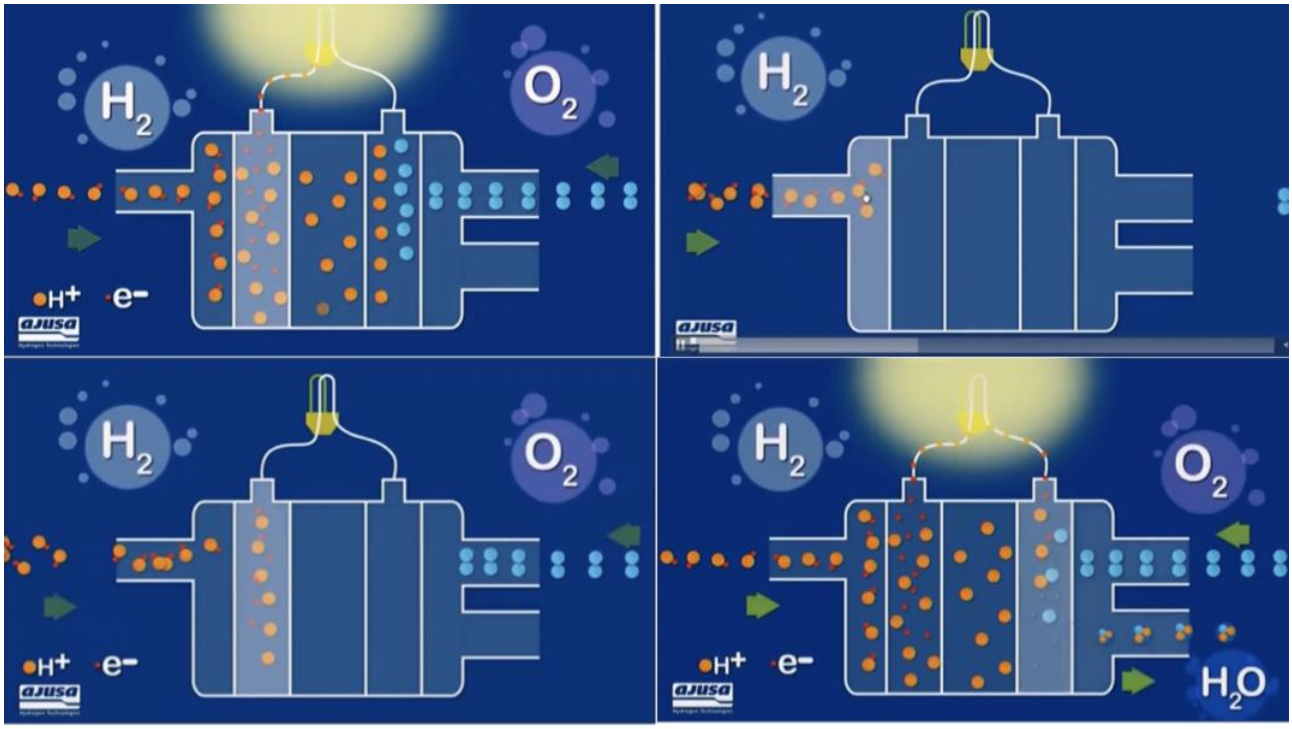

Figure 20: Electrolysis

There are three possible electrolyzers. "Alkaline electrolysis is a mature and commercial technology. It has been used since the 1920s, in particular for hydrogen production in the fertilizer and chlorine industries. Several alkaline electrolyzers with a capacity up to 165 megawatts electrical (MW_e) were built in the last century in countries with large hydropower resources (Canada, Egypt, India, Norway and Zimbabwe), although almost all of them were decommissioned when natural gas and steam methane reforming for hydrogen production took off in the 1970s. Alkaline electrolysis is characterized by relatively low capital costs compared to other electrolyzer technologies due to the avoidance of precious materials. PEM electrolyzer systems were first introduced in the 1960s by General Electric to overcome some of the operational drawbacks of alkaline electrolyzers. They use pure water as an electrolyte solution, and therefore they avoid the recovery and recycling of the potassium hydroxide electrolyte solution that is necessary with alkaline electrolyzers. They are relatively small, making them potentially more attractive than alkaline electrolyzers in dense urban areas. They are able to produce highly compressed hydrogen for decentralized production and storage at refueling stations (30–60 bar without an additional compressor and up to 100–200 bar in some systems, compared to 1–30 bar for alkaline electrolyzers) and offer flexible operation, including the capability to provide frequency reserve and other grid services. Their operating range can go from zero load to 160% of design capacity (so it is possible to overload the electrolyzer for some time, if the plant and power electronics have been designed accordingly). Against this, however, they need expensive electrode catalysts (platinum, iridium) and membrane materials, and their lifetime is currently shorter than that of alkaline electrolyzers. Their overall costs are currently higher than those of alkaline electrolyzers, and they are less widely deployed. SOECs is a new electrolysis technology. They have not yet been commercialized, although individual companies are now aiming to bring them to market. SOECs use ceramics as the electrolyte and have low material costs. They operate at high temperatures and with a high electrical efficiency. Because they use steam for electrolysis, they need a heat source. If the hydrogen produced were to be used for the production of synthetic hydrocarbons (power-to-liquid and power-to-gas), the waste heat from these synthesis processes (e.g. Fischer-Tropsch synthesis, methanation) could be recovered to produce steam for further SOEC electrolysis. Nuclear power plants, solar thermal or geothermal heat systems could also be heat sources for high-

temperature electrolysis. Unlike alkaline and PEM electrolyzers, it is possible to operate an SOEC electrolyzer in reverse mode as a fuel cell, converting hydrogen back into electricity, which means it could provide balancing services to the grid in combination with hydrogen storage facilities. This would increase the overall utilization rate of the equipment. It is also possible to use a SOEC electrolyzer for co-electrolysis of steam and carbon dioxide, producing a gas mixture (carbon monoxide and hydrogen) for subsequent conversion to a synthetic fuel. One key challenge for those developing SOEC electrolyzers is addressing the degradation of materials that results from the high operating temperatures". [20] Hydrogen can be produced through the SMR method (about half of the hydrogen produced currently involves the use of this technique, using natural gas) from biomass or through electrolysis (among which alkaline electrolysis is the cheapest and technically most mature).

Applicazione	Capacità energetica	Efficienza energetica	Costi di investimento	Vita	Maturità
SMR grande scala	150-300MW	70-85%	290-435€/kW	30 anni	Maturo
SMR piccolo scala	0.15-15MW	51%	2,175-3,626€/kW	15 anni	Introduzione sul mercato
Elettrolizzatori alcalini	Fino a 150MW	65-82% (HHV)	616-1088€/kW	60000-90000 ore	Maturo
Elettrolizzatori PEM	Fino a 150kW (stacks), fino a 1MW (sistema)	65-78% (HHV)	1008-2755€/kW	20000-60000 ore	Introduzione sul mercato
Elettrolizzatori SO	Scala di laboratorio	85-90% (HHV)	-	10000 ore	R&D

Table 3: Hydrogen production methods

Fossil fuels need to be replaced not only for energy production but also in chemical production. According to BP data from 2019, oil reserves from 1998 to 2018 are growing, however there is no international criterion to evaluate what oil reserves are, but considering them correct, in 2018 the reserves amounted to 1,729.7 billion. barrels and with these numbers we would have around 48 years of consumption at current levels. If we consider in this calculation what are called URRs, that is, those resources that we know exist but technically cannot find them in their current state, there are at most 4 times the number of years expected with the use of oil alone. There are 2 types of oil, the so-called "easy" one, namely that of Saudi Arabia or the Persian Gulf, in which by drilling the wells the oil comes out spontaneously and then there is the extreme or unconventional oil that is obtained for example from the sands bituminous or through fracking, activities that require a huge expenditure of energy, economic but above all environmental. The sooner we decrease our reliance on fossil fuels and develop new energy sources, the better. The demand for lithium-ion battery technology is simply growing faster than the supply of lithium can satisfy. So, it seems clear: we need a multi-faceted approach to solve this problem. A solution to take over from fossil fuels is the usage of the hydrogen and company like Toyota and Shell are working to develop this industry. It won't be an easy race. Hydrogen has three primary obstacles it needs to overcome to become a viable energy source for any industry. Safety, infrastructure and cost. Hydrogen has a massive advantage over oil-derived fuels, it is lighter than air, it can be purged using emergency valves in the event of fire. The electric grid is a pre-built transportation and generation network for the fuel battery- operated vehicles require and installing a charger in your driveway or garage isn't a huge challenge. Hydrogen does not have such luxuries to kick start the hydrogen economy. There are few large-scale production facilities in the world, with the largest being Shell's Rhineland oil refinement facility, it uses its own hydrogen production in the oil refinement process. Shell opened the UK's first ever hydrogen fuel pump, but the question at this point is: how the hydrogen got there? Transporting hydrogen in pressurized trucks would be too expensive as no large-scale production facilities nearby and hydrogen cannot be transported within the already established natural gas pipelines around the world, so Shell and ITM took the next logical step to keep cost down: they built a hydrogen production and storage on site. The production facility is placed just behind the main station and it is capable of producing 80 kilograms of hydrogen a day.

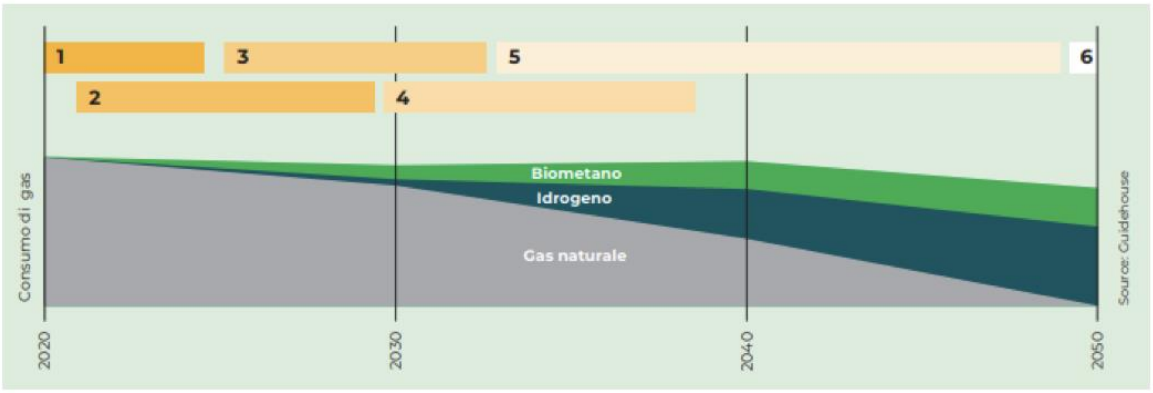

Figure 21: Evaluation of Hydrogen, Biomethane and Natural Gas until 2050

It is estimated that in 2030 the annual fuel expenditure of a small car will be about 353 euros lower than the cost in 2020, thanks to the efficiency of electric cars. GDP will be 2396 million euros higher than today, thanks also to the reduction in oil imports and the creation of new jobs. Carbon dioxide emissions will reduce nitrogen monoxide emissions by 32% and by 65% compared to the levels recorded in 2017 with an even greater percentage in the decrease in particulate emissions, estimating that a large number of cancers will be avoided by 2030 lung and chronic bronchitis avoiding more than 1000 premature deaths [21]. The development of the electricity and hydrogen market would also bring advantages from an economic point of view, being able to produce electricity and hydrogen in a renewable way by renouncing oil imports which from the Petroleum Union data to 2017 amounted to about sixteen millions of tons of refined petroleum products. The objectives outlined by Europe in order to decrease dependence on fossil fuels with the search to improve air quality by reducing emissions of climate-altering substances can be summarized in certain fundamental actions: decrease private transport by reshaping the modal split and shifting the demand for mobility towards public transport, making it more efficient and accessible and improving technologies related to mobility, through the diffusion of alternative and electric fuels in all areas of road transport.

1.6 Normative context

The policy maker represented by the Public Governance pursue social and sustainability goals, encouraging new mobility services in a sharing and green scenario, also through the definition of rules. Regarding ITS, The European Union has considered these systems to be strategic by publishing Directive 40 of 2010 (6 August 2010) which entered into force on 26/08/10 which aims to create a basis for the use and dissemination of these systems in a coherent and coordinated way in the various states of the European Union with particular attention to the borders of the member states but above all to establish for each state the general conditions necessary for this development. In Italy, the European directive was transposed under the Decree-law of 18/02/12, article 8 entitled "Measures for the innovation of transport systems". The decree was converted into the Law in December 2012 and with the Decree of February 1, 2013 on the “Diffusione dei sistemi di trasporto intelligenti in Italia”. The decree identifies the intervention sectors, such as mobility, freight transport, traffic data management, to increase road safety and finally the telematic link between vehicles and transport infrastructure and as stated in article 3 there are many requirements for the diffusion of ITS systems: to clearly contribute to the problems of mobility, i.e. traffic, congestion, reduction of pollutants and increase road safety by ensuring intermodality and promoting equal access to ITS services and applications by also developing , on-board vehicle technologies to improve communication by promoting V2V. The fact that the transposition took place in the context of the Development Decree Bis is a recognition that the transport sector and in particular that ITS can be a fundamental tool for the growth of the country, therefore, it is useful not only for the improvement of some situations of mobility but in general for the growth of the country. The issues of 2013 were taken up in the National Action Plan on Intelligent Transport Systems of February 2014.
"Finally, the European Commission has published four Delegated Regulations, which integrate Directive 2010/40/EU and which therefore constitute community rules to be respected when, as happened, Italy has implemented Directive 2010/40 / EU. These Delegated Regulations are:
• Commission Delegated Regulation (EU) no. 305/2013 of 26 November 2012 which integrates Directive 2010/40/EU of the European Parliament and of the Council with regard to the harmonized preparation throughout the territory of the European Union of an interoperable electronic emergency call service (eCall), published in GUCE on April 3, 2013.
• Commission Delegated Regulation (EU) no. 885/2013 of 15 May 2013 which integrates Directive 2010/40/EU of the European Parliament and of the Council with regard to the provision of information services for secure parking areas for heavy vehicles and commercial vehicles, published in GUCE on 18 September 2013.
• Commission Delegated Regulation (EU) no. 886/2013 of 15 May 2013 which supplements Directive 2010/40/EU of the European Parliament and of the Council with regard to data and procedures for the provision, where possible, of free universal minimum traffic information for road safety, published in GUCE on September 18, 2013.
• Commission Delegated Regulation (EU) no. 962/2015 of 18 December 2014 which supplements Directive 2010/40/EU of the European Parliament and of the Council relating to the provision throughout the European Union of traffic information services in real time, published in the Official Journal on 23 June 2015 ". [22]
In addition to the ITS diffusion, the development of new sustainable “fuels” is of paramount importance to achieve the climatic goals, decarbonization and independence from fossil fuels, and in particular the hydrogen plays an important role in the industry and mobility scenario.
The European Commission sees hydrogen as an opportunity for the energy transition, for the decarbonization of various sectors. Currently, however, hydrogen is relegated to a few sectors of industry and is produced mainly from fossil sources, more than 90% of the hydrogen currently produced in Europe comes from fossil fuels, only 1.6% is produced by electrolysis. Europe sees in the hydrogen energy vector an opportunity not only in the industrial field but also in the mobility field, for which it has placed a series of time horizons within the "hydrogen strategy" published

8/07/20 (2024, 2030, 2050) for the development of hydrogen at European level in order to develop a strong and cohesive supply chain at European level; has set objectives in the production of hydrogen with low CO2 emissions by setting as a goal the installation of electrolyzers (40GW by 2030), this will lead to a consistent development of hydrogen up to 2050 so that hydrogen becomes a quarter of consumption energy at European level. A tool used by the European Commission is the Clean Hydrogen Alliance, an alliance of companies, public authorities, research centers, financial institutions to support the development of concrete projects at European level, it is divided into roundtables from production to final use in mobility, in the energy and residential sectors.

Figure 22: European and Italian strategy for green policy

"The Directive 2014/94 / EU of the European Parliament and of the Council, of 22 October 2014, on the construction of an infrastructure for alternative fuels (DAFI) was implemented in our legal system with Legislative Decree no. 16 December 2016, no. 257. With the aim of minimizing dependence on oil and mitigating the environmental impact in the transport sector, it establishes:

• the minimum requirements for the construction of the infrastructure for alternative fuels, including charging points for electric vehicles and refueling points for natural gas (LNG and CNG) and hydrogen, to be implemented through the National Strategic Frameworks of the Member States;

• common technical specifications for these recharging and refueling points, and requirements concerning information to users.

Europe wants to reach climate neutrality, by 2050, by developing the Green Deal in which a fundamental role is attributed to the use of hydrogen to achieve the decarbonization of various sectors of industry. Therefore, one of the objectives is also to decrease dependence on oil and reduce the climate impact, especially in vehicular mobility. The Green Deal aims to improve the quality of life, in terms of pollution, bringing mobility and industry towards a radical green transformation with zero emissions and in economic terms which, as President Ursula von der Leyen said, is a strategy for growth, to reduce emissions and create jobs. The aim of the directive is the development of a large market for alternative transport fuels, which are identified in: electricity, natural gas and hydrogen. Each type of propellant is subject to a regulatory provision relating to its distribution ". [23]

Chapter 2: Case study

2.1 Case Study – FCHEV

The center of our experimentation was the FCHEV developed by CNR ITAE of Messina. The concept behind this vehicle is to overcome the problems related with strictly electric or hydrogen powered vehicle giving a solution that is capable of giving different advantages in terms of efficiency from different point of view.
The idea of this thesis is to create a platform that can be installed directly on board the vehicle, a system capable of monitoring and sending (soon) to the server or to other destinations (other vehicles or infrastructures) the data deductible from the sensors connected to the platform and data from the CAN bus with which the platform communicates. All this is useful in a communication and interaction scenario between vehicles but even before to acquire and represent data, to implement predictive maintenance, to carry out assessments, for example, of the need for recharge.
The prototype of a bus with electric fuel cell and battery hybrid propulsion is suitable for the public transport service class B - category M3. The main components of the powertrain and the data exchanged via CAN messages inside the vehicle will be examined.
For the realization of the prototype of a bus with electric fuel cell and battery hybrid propulsion, a long-cantilevered IVECO minibus model Way was used with the following characteristics:

Fuel cell hybrid electric minibus data sheet	
Vehicle platform	IVECO DAILY
Dimension (length x width x height)	7348 x 1996 x 3100 [mm]
Laden mass	5600 [kg]
Seating capacity	16 + 1 [B Class on Italian Regulation]
Traction	Asynchronous electric engine 3 phase AC with IGBT inverter
Nominal engine power	40 [kW]
Power peak	80 [kW] @ 2950 [rpm]
Battery	Li-Ion polymer
Battery energy	70[kWh]
Battery nominal power	30[kW]
Battery peak power	120[kW]
Fuel cell type	PEM
Fuel cell system power	20[kW]
Hydrogen storage	300 [liters] @ 350 [bar]
Consumption (average)	0.75 [kWh/km]
Range FCHEV	>240 km
Range BEV	>100km

Table 4: Data Sheet CNR ITAE FCHEV

Figure 23: CNR ITAE FCHEV

The main features are the presence of an electric motor, a traction inverter, a traction battery pack and a fuel cell system suitably sized according to the chosen usage strategy.
The FCS has been associated with traction batteries to increase the autonomy of the vehicle. This distributes the electrical power to the connection line between the batteries and the motor-inverter unit via a DC / DC converter.

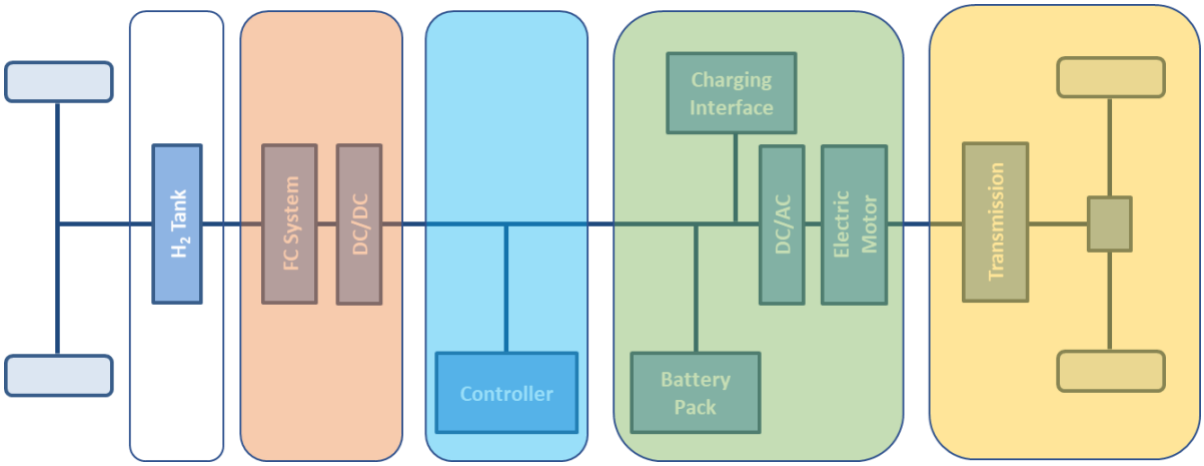

Figure 24: FCHEV Power Train

The Fuel Cell system works at constant power and we can distinguish two operating phases highlighted in Figure 25, taken from the article “Development of a fuel cell Hybrid electric powertrain: A real case study on a Minibus application”: if the engine has a demand for power greater than that the fuel cell system delivers, the fuel cell system itself supplies power to the battery to fill the gap and make possible the movement of the vehicle, if the engine has a demand for power lower than that the FCS delivers, the fuel cell system charge the battery pack.

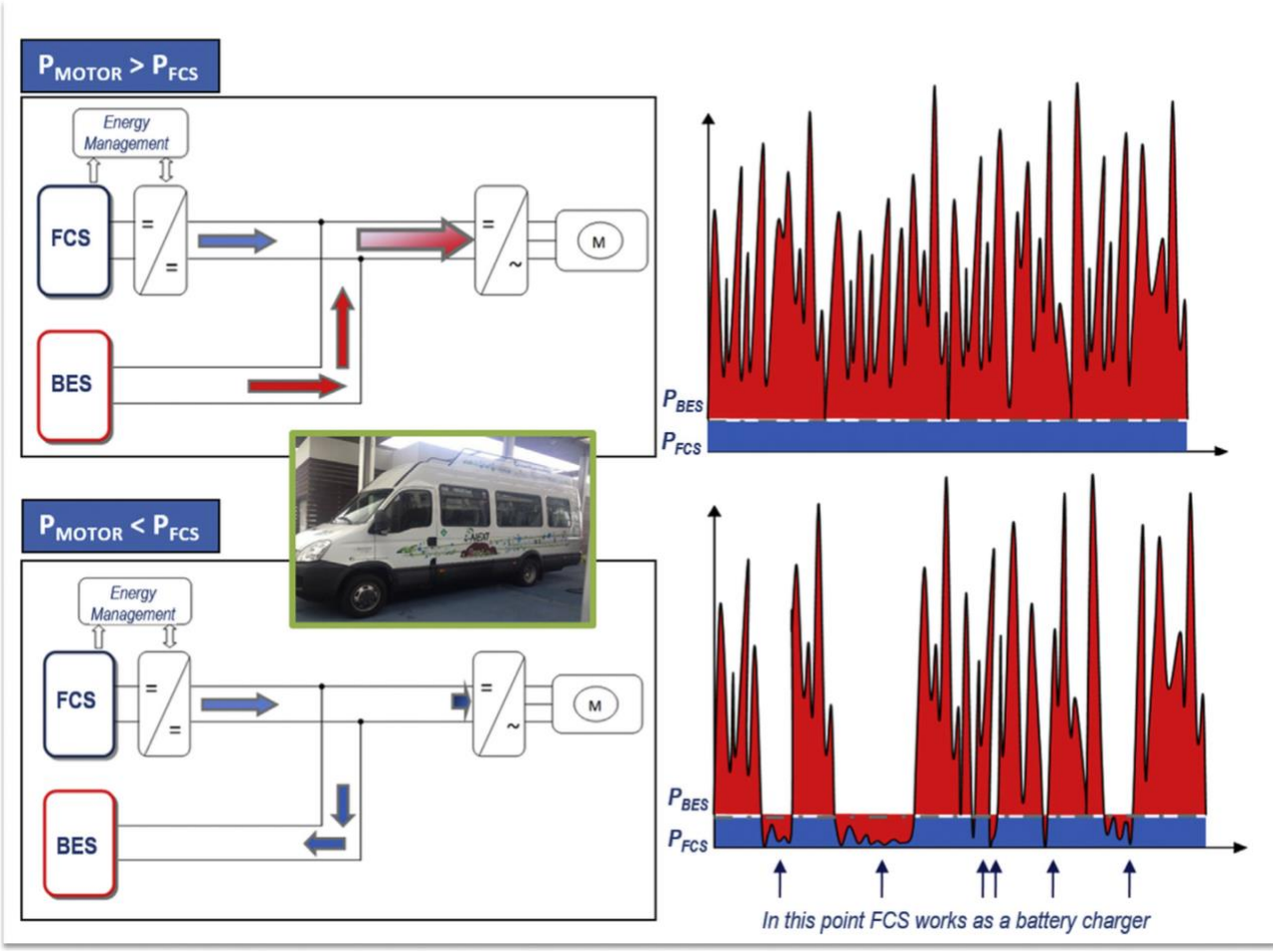

Figure 25: Operating Phases FCHEV [24]

The activation logic of the Fuel Cell depends on the SOC of the batteries, in particular it is envisaged that the Fuel Cell starts operating as soon as the SOC is below the minimum set threshold and it switches off when the SOC reaches the maximum set threshold. This logic increases the life time of the Fuel Cell system because by working at constant power, sudden on and off phases are avoided, also increasing the useful life of the batteries because they are subjected to less stressful charge / discharge cycles compared to only electric vehicles. The activation and deactivation of the Fuel Cell system is described for the following phases: Phase 1, the vehicle moves and the FCS is deactivated, Phase 2, the Fuel Cell system is activated to reach the minimum threshold and recharges the battery, Phase 3, the Fuel Cell is deactivated when the maximum threshold is reached, Phase 4, vehicle in motion and the Fuel Cell system is active and the minimum threshold is reached.

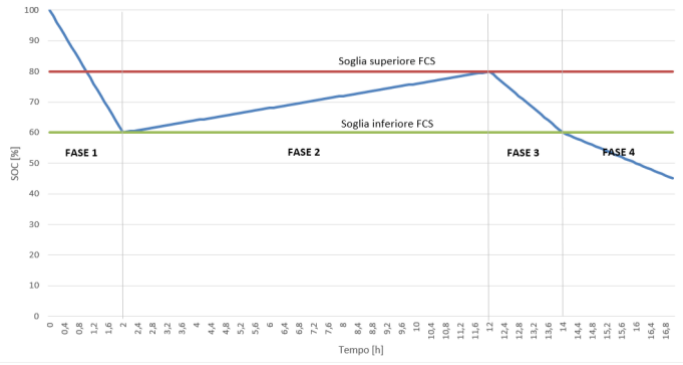

Figure 26: Activation Threshold FCS

The energy storage system (traction battery pack) has an energy capacity of approximately 70 kWh, which allows the vehicle to travel a distance greater than 100 km in BEV (Battery Electric Vehicle) mode and more than 240km in FCHEV mode (Fuel Cell Hybrid Electric Vehicle). This solution has the advantage of optimizing regenerative cycles by reducing consumption and maximizing the total range of the vehicle.

The electrical architecture consists of three main lines:

- 280-400 Vdc high voltage line;
- 70-120 Vdc medium voltage line;
- 12 Vdc low voltage line.

The first two lines represent the power plant intended for the generation, transformation and use of energy for vehicle traction. The high voltage line connects the batteries to the traction system and to the power auxiliaries (inverters and compressor motors). The medium voltage line transfers the power generated by the Fuel Cell to the high voltage line by means of a voltage converter (DC-DC). Finally, the low voltage line is used to power all the vehicle auxiliaries and is connected to a 12V battery which is recharged via a DC-DC connected to the high voltage line.

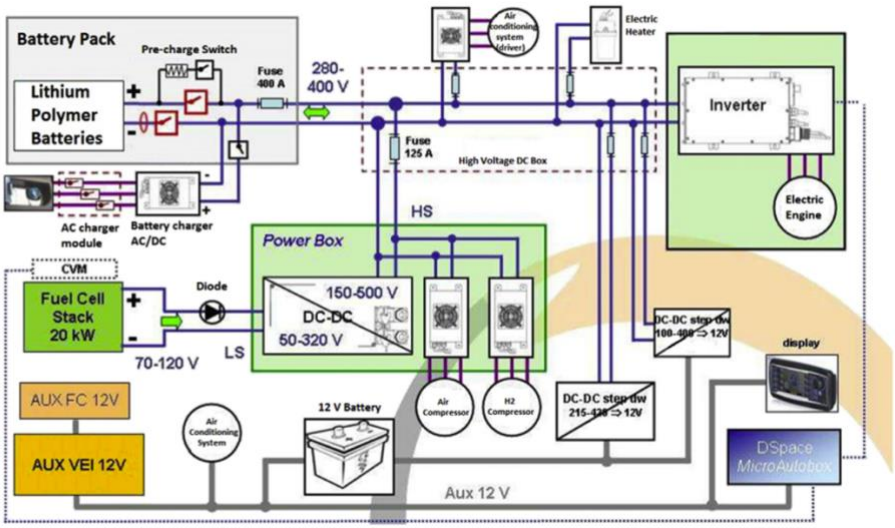

Figure 27: Functional electric scheme of FCHEV [25]

The interface between the control unit and the components of the minibus takes place through four CAN lines:

- CAN B @ 50 kbps;
- CAN CLUSTER (NQS) @ 50 kbps;
- CAN VEI @ 250 kbps;

• CAN FCS @ 500 kbps.

The first CAN line (CAN B) interfaces the body computer with the "instrument panel node" (NQS), between these systems a control unit has been interposed, capable of intercepting some CAN messages from the original vehicle and converting them into useful data for the new vehicle. In this way, for example, the "fuel level" information becomes "hydrogen pressure level", the engine revolutions become the outgoing / incoming power from the battery pack, and so on. Furthermore, some warning / fault signals present on the original vehicle have not been reused (such as the ignition of the glow plugs or the engine oil level) because they were deemed useless. The CAN cluster or CAN NQS is the CAN B line "filtered" by the control unit to which some information is added. The CAN VEI is used to interface the control unit to the vehicle, in particular for reading the signals useful for the control and for the messages to be sent to the instrument panel node. Furthermore, on this line there are the vehicle control units, which communicate according to the same j1939 protocol at 250kbps. The CAN nodes that are part of the Fuel Cell system are connected to the CAN FCS, in particular air compressor inverter, hydrogen recirculation compressor inverter, DC-DC power (high voltage) between medium voltage line and high voltage line, traction inverter, and battery management unit (BMU).

The MES-DEA 200-330 series electric traction motor, rotostator unit with a diameter of 200 mm and a length of 330 mm of active part, with the following characteristics:

• maximum continuous power 40 kW;

• peak power of 80 kW;

• base speed 2950 rpm;

• operating voltage 250 V;

• continuous torque 130 Nm;

• maximum speed of the electric motor: 10,000 rpm.

The MES-DEA electric traction motor is firmly coupled to a speed reducer.

Riduttore di velocità	
Tipologia	Single reduction
Rapporto di riduzione [Nm]	1:3,93; 300Nm
Coppia massima [Nm]	300Nm
Lubrificazione	Olio
Raffreddamento	Ad aria

Table 5: Speed Reducer Characteristics

An HS (hard switching) inverter of the MES-DEA model TIM600W was chosen for the powertrain characterized by 600 V and 600 A peak IGBT modules. In particular, the maximum operating voltages are 400 V full power (full current) while the maximum phase currents that can be generated reach 400A rms. The TIM600W is an inverter capable of driving, in real operating conditions, electrical machines delivering mechanical powers up to 80 kW, therefore it has a safety coefficient suitable for the performance that the minibus must express.

The main features are listed below:

• input voltage: 80-400 Vdc;

• nominal output current: 235 Amps;

• max output current: 350 Amps;

• control: PWM;

• communication: CAN.

The battery pack, shown in Figure 29 is located under the floor of the minibus in the advanced central area between the front axle and the electric motor. It consists of 16 modules with Lithium Ion Polymer technology from the Korean company KOKAM. Each module consists of 6 cells with a capacity of 200 Ah. In total, the battery pack has a total rated voltage of 355 V and an energy of 70 kWh. The battery pack was preliminarily tested in discharge to verify both the performance of the cells, evaluating the voltage as the drained current varies, and the information transmitted on the CAN bus from the BMU and subsequently the vehicle was prepared for their installation. The hydrogen storage on board the minibus consists of two Dynetek cylinders shown in Figure 28, of 150 liters and a capacity of approximately 3.5 kg of hydrogen at 350 bar per cylinder. These were installed on the minibus roof, taking advantage of the available roof surface and using a suitably designed support frame.

Figure 28: Hydrogen Tank

Figure 29: Battery Pack [24]

The Fuel Cell System shown in Figure 30 and schematically in Figure 31 (taken from the article "Study and design of a hybrid electric vehicle", L. Andaloro, A. Arista, G. Agnello, G. Napoli, F.Sergi, V. Antonucci, International Journal of Hydrogen Energy, Volume 42, Issue 5, 2 February 2017, Pages 3166-3184, Elsevier) was installed under the rear floor inside a protective box in the former spare wheel compartment in order to protect it from any impacts. The stack was installed in an advanced central position, then the air line (cathode side) was assembled, including filter, compressor and stack inlet and outlet shut-off valves; the low pressure anode line (0.5 bar) including recirculator; the stack cooling line including deionized water pump and plate heat exchanger interfacing with the cooling system installed on board the vehicle.

Figure 30: Fuel Cell

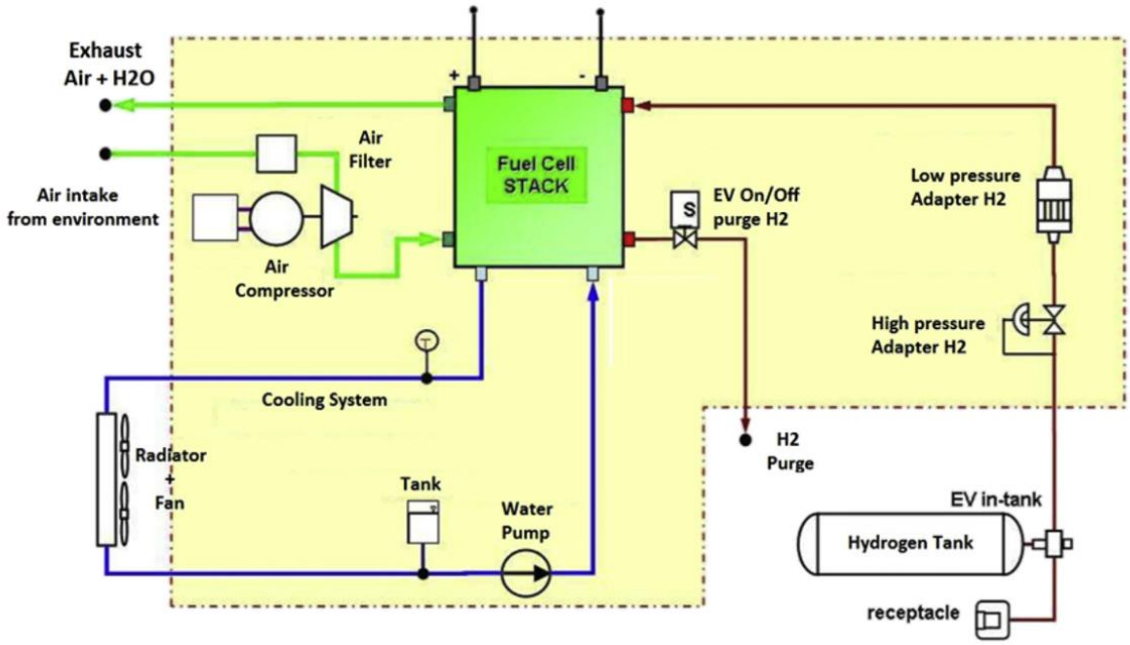

Figure 31: Fuel Cell System [25]

2.2 Capo d'Orlando experimental site

The green production of the hydrogen is crucial to achieve the climate objectives that the world is indicating, in this context the I-Next (innovation for green Energy and eXchange in Transportation) project takes place in the Capo d'Orlando experimental site (summarized in figure 32, taken from "Electrochemical energy storage mitigating impact of electric vehicle on the electric grid: two Italian case studies"), where the CNR FCHEV is located. This is one the first Italian solar-powered hydrogen production and fueling plant developed by CNR-ITAE of Messina and other industrial partners. The plant is mainly divided into four sections: hydrogen production and purification module (HPPS), a hydrogen compression module (HCS), a hydrogen storage facility (HSS) and a hydrogen refueling facility (HRS). All the different phases related to the compression, production and refueling are managed through a PLC control panel. The production module is composed by different elements, a deionization water system, an alkaline electrolyzer (produced by ErreDue Spa) and a deoxidation and drying gas system. Once the electrolyzer, that is a part of the system highlighted in Figure 33, taken from the article "Freight distribution with electric vehicles: A case study in Sicily. RES, infrastructures and vehicle routing", produces hydrogen starting from the water, the hydrogen is sent to the deoxidation and drying gas system where it is purified. This plant is totally powered by a renewable energy source, that is the sun, through a set of photovoltaic modules (420) that are placed on the roof of UDC shed of 720 square meters shown in Figure 34. The electricity produced in this way is able to supply both on-site hydrogen plant and an electric charging station for BEV, FCEV and FCHEV. "The peculiar characteristic of this plant is that , since it is connected to a photovoltaic plant and a Battery Energy Storage System (BESS), the electrolyzer is able to generate hydrogen regardless, too, of the presence of the renewable source or refueling demand, also the entire system is grid connected which could facilitate their operation in electricity markets.[27]" A scheme of the production plant is showed in Figure 32 (Electrochemical storage mi mitigating impact of electric vehicle on the electric grid: two Italian case studies, Ferraro M., Andaloro L., Sergi F., Aloisio D., Dispenza G., Napoli G., Micari S., Brunaccini G., Randazzo N., Di Novo S., Antonucci V.)

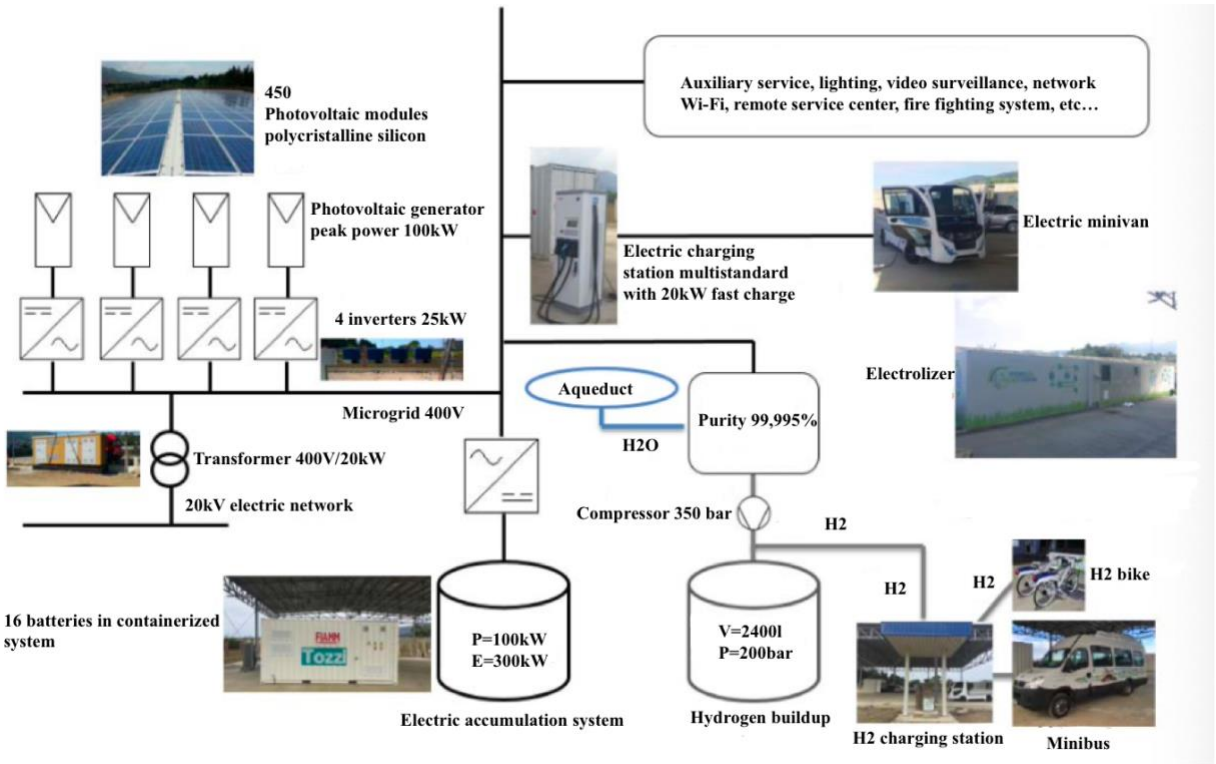

Figure 32: Capo D'Orlando I-Next Plant [26]

Figure 33: Hydrogen Filling Station [24]

Figure 34: Photovoltaic modules [24]

Chapter 3: Development of the communication platform

As introduced in chapter 2, the aim of the thesis is to show, in the new connected and eco-sustainable mobility scenario, the possibilities offered by a platform capable of acquiring different types of data and managing them. In figure 35 is showed the conceptual overview. The figure shows the interaction between the vehicle and a microcontroller, through which we can analyze different parameters, exploiting environmental sensor, localization sensor, CAN interface to be able to receive all the messages coming from the control units aboard the vehicle; once all the needed parameters are acquired and stored in a sim card the communication between the micro and the server is enabled to observe and to organize in a proper way the data. Two possible future implementations are V2V and V2I.

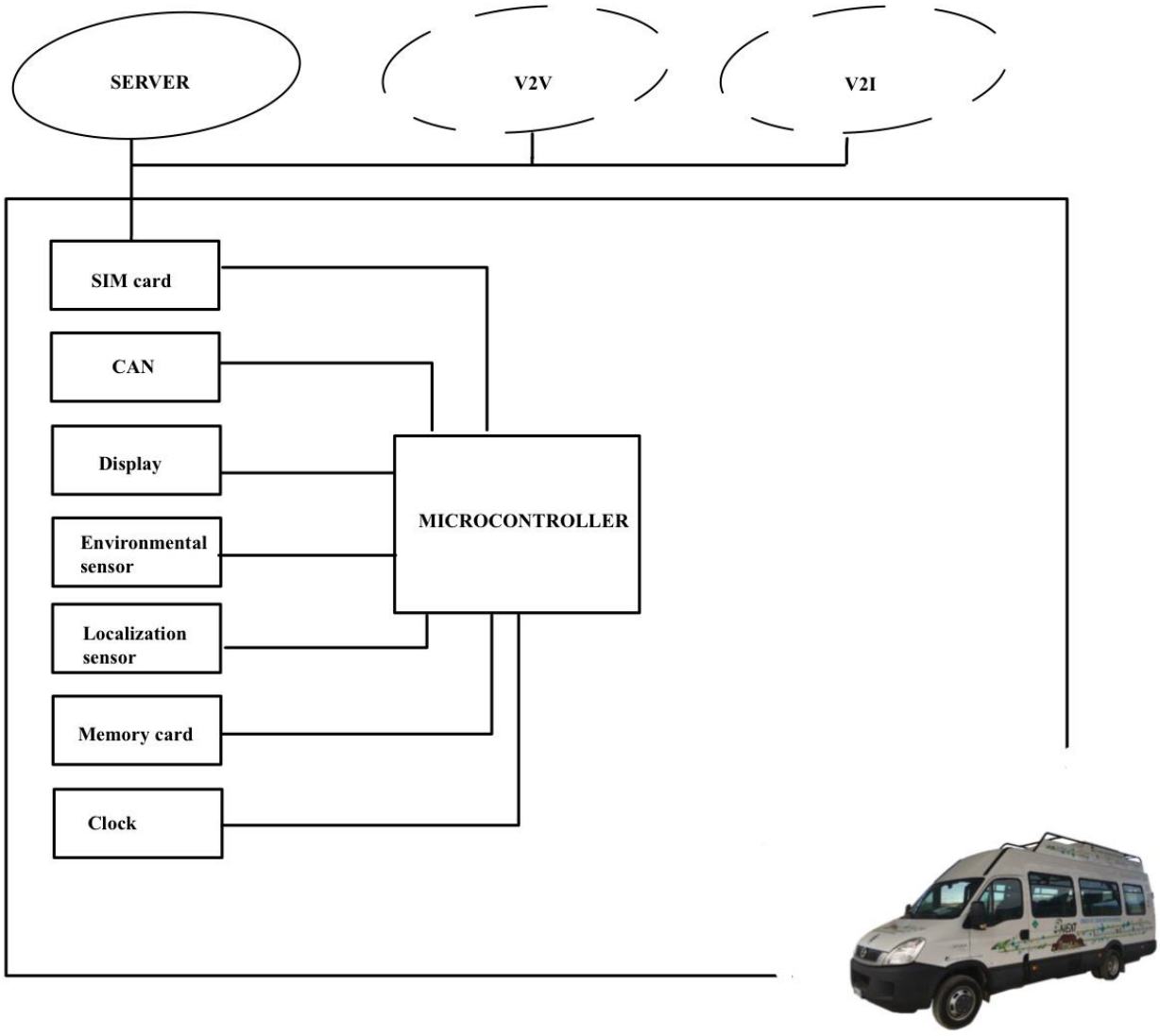

Figure 35: Overview of the whole project

The project developed in this thesis makes use of a microcontroller called STM32F4ZI and the subsystems interfaced with this micro are an RTC, an OLED display, a BMP180, that is a pressure, temperature, altitude sensor, a GPS sensor for locating the vehicle and an SD card to store the data from these sensors. A conceptual scheme is showed in Figure 36.

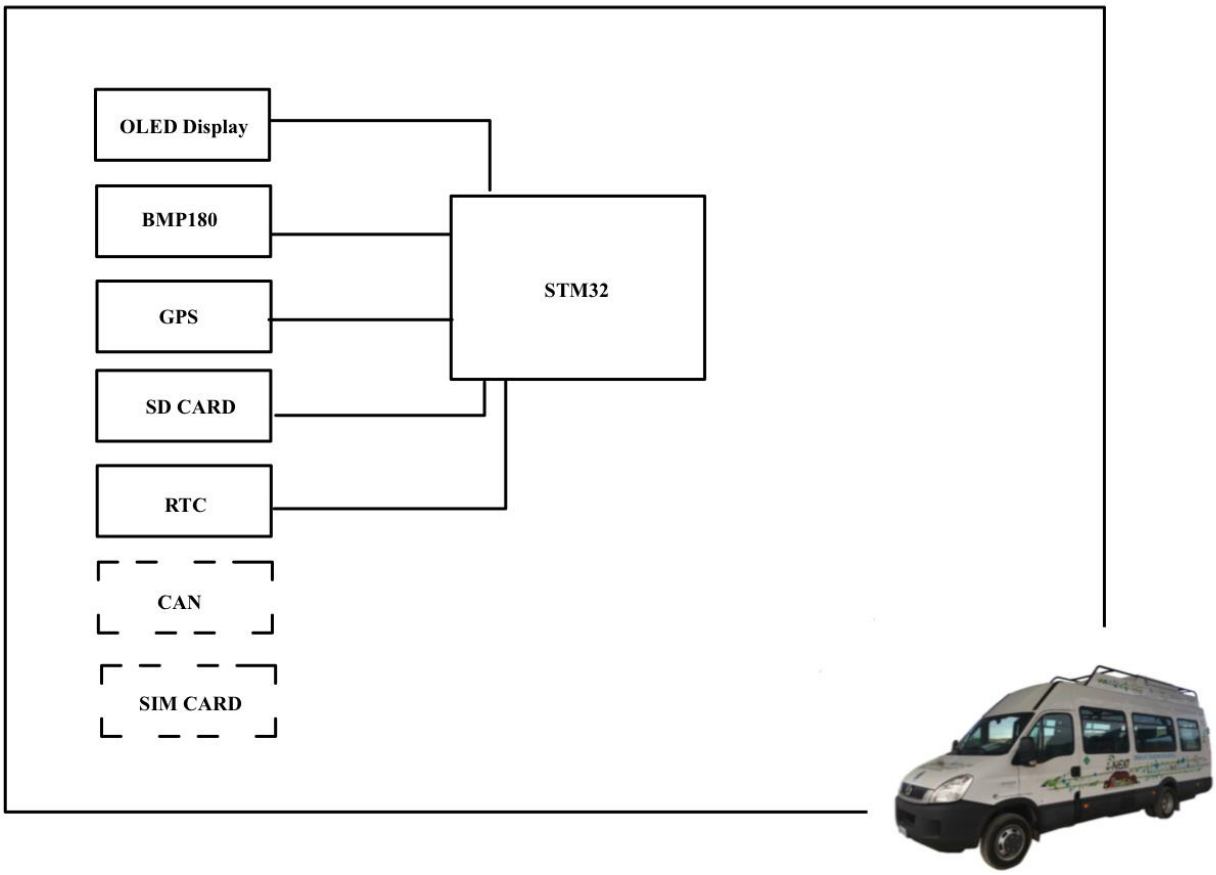

Figure 36: Overview of the thesis work

The following Section 3.1 and 3.2 describe the development of a single board based on STM32 ecosystem. In addition, communication with the CAN bus and GSM shield was carried out using an Arduino device, through a totally-based ARDUINO system described in the following Chapter 4.

3.1 Communication protocols

Considering the ITS context depicted in Chapter 1- the new possibilities offered by communication technologies applied to automotive industry and the importance of such strategies to achieve essential social, economic and climatic goals- and the description of the vehicle in Chapter 2- a fuel cell hybrid vehicle that makes use of "green hydrogen" directly produced on site in Capo D'Orlando I-Next Center- the aim of this section is to show the potentialities offered by the interaction between the vehicle, different devices and a system able to interact with all these subsystems exploiting different type of communication technology, in such a way that all these devices can interact with each other in a non-overlapping way (the state machine is not developed in this thesis). The adopted microcontroller (the system that takes care of interfacing with other devices to provide some output to the final user) is an STM32F429ZI, a very versatile and highly configurable system. The choice of such microcontroller depends on the performance that it ensures compared to ARDUINO, this last one uses an ATMega microprocessor that is lower in performance and features than an STM32 one; ARDUINO platform is quite easy to configure, this advantage makes these systems more widespread than STM32, but the choice of this last one has been made for its performance although it is less used and more complex than ARDUINO. The chosen microcontroller can interact with different devices to acquire information related to the time, pressure, temperature, altitude, position and vehicle sensor parameters. In a future scenario, not developed in this thesis, all these parameters will be helpful in the communication environment from a vehicle to another vehicle or to an infrastructure or to a pedestrian exploiting these data in real-time to take crucial decisions. (To see the code of each component in detail go to the Appendix).
Embedded systems are computers and computers are based on CPU that is responsible for running software, in order to be able to perform that execution CPU needs to be able to access some peripherals that are the memory, that contains the instructions as well as the data that we need to process in order to execute our application and input/output devices that are elements through which we can interact with the application. The operations are initiated by the CPU, when it needs to read some data from memory a read cycle is performed, otherwise when it needs to send some data to memory or input/output devices a write cycle is performed; all the resource that are accessible by a CPU are associated with an address that is a unique binary configuration that univocally identify the resource the processor is interested to. There are different implementations for an embedded system hardware, for example the microcontroller-based implementation where a single device hosts most of the components, CPU, RAM memory, Flash memory, this configuration is shown in figure 37 Massimo Violante, Operating systems for embedded systems).

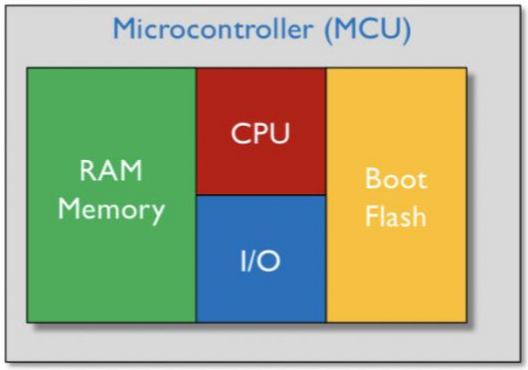

Figure 37: Microcontroller [25]

We have 2 possible memory solutions: non-volatile and volatile memory. A non-volatile memory is used to store programs to be executed at power-up and it is characterized by a slow access, the volatile one is not able to retain the content when the computer is powered off but it's very fast after power-

up is completed; the RAM memory is a volatile memory and Flash memory is non-volatile one. We have to remember that when dealing with a microcontroller we are always poor in terms of resources, so we have to find a way to optimize.

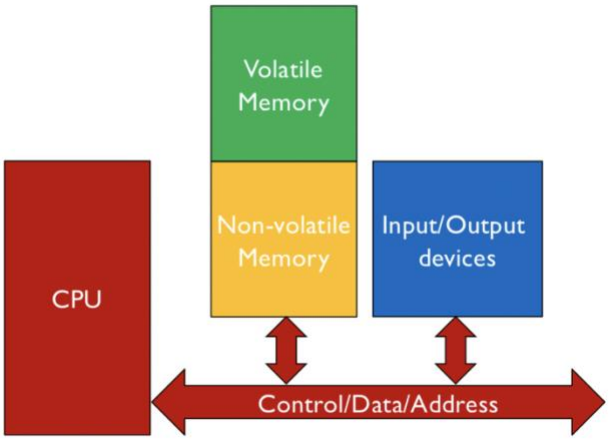

Figure 38: Embedded systems

Memory and input-output devices are connected with CPU through a bus (as shown in figure 38, Massimo Violante, Operating systems for embedded systems), a collection of lines that transfer information from the CPU to memory and input/output devices and vice versa, typically the bus is composed by 3 kind of information: control information, data information and address information. We have mainly two options when interfacing the CPU and a peripheral, parallel and serial bus. In the parallel bus configuration there is one physical line for each bit of information that we want to transfer, the disadvantage here is the high pin count, we have to allocate enough rooms for all these interconnections and we can have the problem of signal integrity, operating at high frequency a line can generate noise in another line due to parasitic coupling exiting between adjacent lines; as a possible solution to reduce the pin count the address and data line can be multiplexed, this means that we obtain a certain number of wires that are shared between different functions and depending on the specific operation at a certain given time, this line can be used to carry address or to carry data. The parallel bus scheme is presented in figure 39 (Massimo Violante, Operating systems for embedded systems).

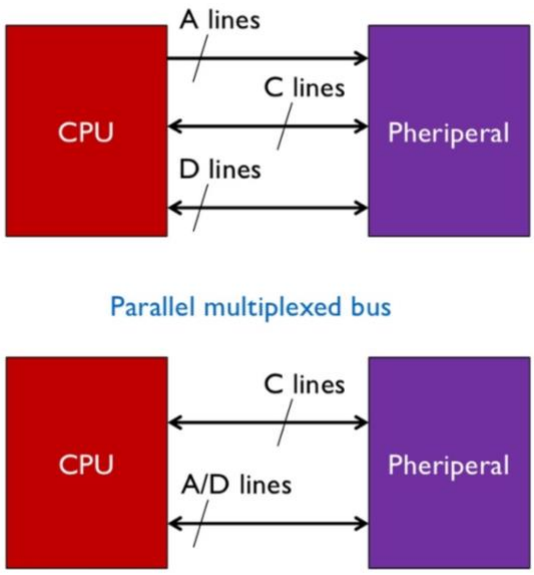

Figure 39: Parallel bus communication[25]

If we adopt a serial configuration, we have a very low pin count because CPU and peripherals are connected with very few lines but as a disadvantage, we have a limited speed. Many protocols are available in the world of serial bus connection, for example SPI, I2C, CAN.

There are a number of interfaces that are used for communication within a computer system. SPI is used for communicating between a peripheral device of some sort and some sort of microcontroller or microprocessor. The interface between the controller and the peripheral is typically a four-wires interface, you always have a clock and the clock signal is always generated by the processor and every bit is transferred upon a clock pulse and we need it because it tells us when a certain operation shall be performed, then you have two data signals, one from the controller to the peripheral device and then one from the peripheral device back to the controller in the other direction and basically the way this interface works is that each time the controller or the master side toggles this clock signal one bit gets sent in each direction and that's roughly how this works. Finally, there is a fourth wire which is a chip select (or slave select) and again this is going from the controller over the peripheral device and the chip select is an active low signal so that needs to be pulled low before any communication will take place and it is a unique line to select each slave chip. The first thing that happens is chip select goes low and then basically at each clock transition from low to high the master is able to send data. The communication happens like this: first I activate the chip select, then I have to send it to zero, then the clock is enabled, then the MOSI and the MISO starts, both the master and the slave start sending their data and the communication ends when the line of chip select comes back high. In the SPI world there is the notion of master and slave, in general, your integrated circuit is going to be your SPI slave that you are talking to and the SPI master is going to be a microcontroller. The SPI master always initiates the transaction sending or receiving of any data and it does so with four signals, the clock, MOSI, MISO and slave select. MOSI means Master Output Slave Input, it goes from the master to all the slaves to transfer data to the slaves, MISO means Master Input Slave Output, it is a wire that connects all the slaves with the master to transfer data to the master.

SPI is full duplex meaning that you can send and receive at the exact same time, you can have data going out MOSI and at the same time data is coming from MISO, so the communications happens in both directions simultaneously. It is faster than UART and I2C but as a disadvantage it requires more pins than UART and I2C.

If the configuration of the SPI is the so called "Daisy chain" one, I have one clock, one MISO, one MOSI and a single chip select that is used all the slaves at the same time, this means that if I want to send a data to a certain slave, for example it is the third of three slaves, I have to send the data first to the slave 1, then it is sent to slave 2 and then slave 3, the slave that needs that data, so the speed of

this method is slower than the one based on the configuration named “Independent slaves” but it has a better usage of the wires. The difference between these two configurations is highlighted in figure 40 (Massimo Violante, Operating systems for embedded systems):

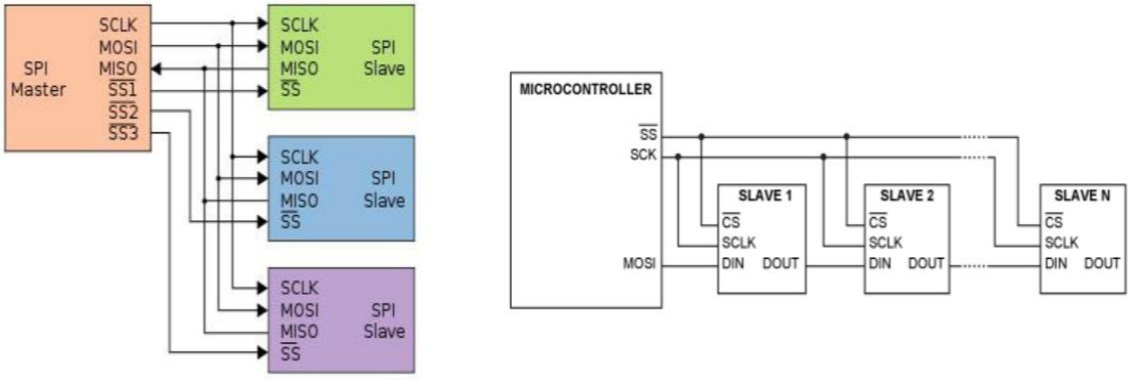

Figure 40: SPI [25]

I2C is a protocol used for sending data back and forth, it is a two-wire interface. At block diagram level you know there is one master and one slave, there are two pull-up resistors and there are two pins, -that makes it extremely attractive from a pin count perspective- the SCL which is the clock and SDA which is your data. Clock is used for aligning the data, so the slave knows when to sample it. We can also have multiple slaves on a single bus which makes it very attractive for interfacing a lot of different integrated circuits that do a lot of different purposes. “The multi-master situations require additional feature of the I2C protocol, the arbitration: arbitration is the procedure by which competing masters decide final control of the bus, I2C arbitration does not corrupt the data transmitted by the prevailing master, arbitration is performed bit by bit until it is uniquely resolved, arbitration is lost by a master when it attempts to assert a high on the data line and fails”. [28]
It shows a moderate transfer speed (up to 3.4 Mbps) but it is very useful for application in which is crucial to reduce the number of wires. In order to initiate communication in the case of I2C we need a start bit and the start bit occurs when the clock is at a high logic value, i.e. one and at the same time there is a switch from top to bottom of the data bus. As for the stop, when the clock is at a high logic value, a switching from the bottom to the top of the data bus must take place. After the start bit, the master takes care of sending the logic value of the clock to zero, at this point the data is sent and then the master returns the logic value of the clock to one and at this point the slave samples the data. The I2C protocol is therefore divided into 7 phases for data transfer, sending the start bit, sending the slave address from the master, sending the write or read bit, sending the ACK bit (acknowledge), sending the data, send ACK bits and send stop bits. In a serial communication, the acknowledge bit is used to make sure that the data has been received / sent correctly.
“A transition of the data line while the clock line is high is defined as either a start or a stop condition, both start and stop conditions are generated by the bus master, the bus is considered busy after a start condition, until a stop condition occurs”. [28]
A scheme of I2C communication is presented in figure 41 (Massimo Violante, Operating systems for embedded systems).

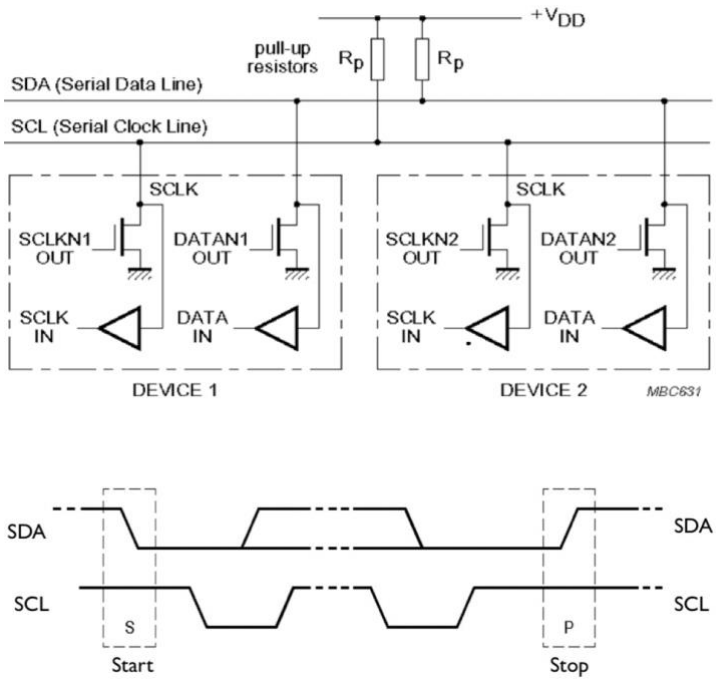

Figure 41: I2C [25]

USART is one the oldest ways to transmit data between two sides through a wire. UART consists of only two wires, that are TX and RX. The USART defines a protocol to exchange serial data between two devices. In the idle state, that is when no data is being transmitted the line is held high, this allows easy detection of damaged line. The start bit is essentially nothing more than a transition from the idle high state to a low state and the user data bits come immediately after the start bit, after the data bits are finished, the stop bit indicates the end of user data, the stop bit is either a transition back to the high or idle state or remaining at the high state for an additional bit time. The data bits are the user data or "useful bits" and they come immediately after the start bit, these data bits are usually transmitted with the least significant bit first.

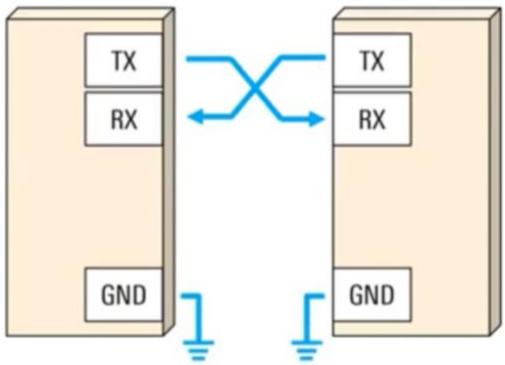

Figure 42: USART

"The UART provides asynchronous communications commonly referred to as RS-232 or RS-485. The UART component can be configured for Full Duplex, Half Duplex, RX only or TX only versions. All versions provide the same basic functionality differing only in the amount of resources utilized.

To assist with processing of the UART receive and transmit data, independent size configurable buffers are provided. The independent circular receive and transmit buffers in SRAM as well as hardware FIFOs help to ensure that data will not be missed while allowing the CPU to spend more time on critical real time tasks rather than servicing the UART. For most use cases the UART can be easily configured by choosing the BAUD rate, parity, number of data bits and number of start bits. The most common configuration for RS-232 is often listed as "8N1" which is shorthand for 8 data bits, no parity and 1 stop bit which is also the default for the UART component. Therefore, in most applications only the BAUD rate must be set. A second common use for UARTs is in multi-drop RS-485 networks. The UART component supports 9-bit addressing mode with hardware address detect, as well as a TX output enable signal to enable the TX transceiver during transmissions".[26] The popularity of UART is decreasing due to other communication protocol as SPI and I2C but it is still relevant for low speed applications. UART consists in a start bit, data bits, parity bit and stop bit. In the idle state the line is high (1), when we have a transition from the idle state to low (0) we have the start bit, the stop bi, instead, is a one. The data bits are the user or useful data and they are typically transferred with the LSB first. "The most common use of the USART in asynchronous mode is to communicate to a PC serial port using the RS-232 protocol". [30]

Figure 43: UART Start and Stop bit

The Controller Area Network, also known as CAN-bus, is a serial standard for field bus (mainly in the automotive environment), of the multicast type. Although initially applied in the automotive field, it is currently used in many industrial applications of the embedded type, where a high level of immunity to disturbances is required. Communication between two devices takes place by exchanging data frames encoded according to a defined structure.
"CAN was developed by Robert Bosch back in 1986 at the request of Mercedes. The need to communicate the many electronic devices inside cars (Antilock Braking System, Traction Control, Air Conditioning Control, central locking, are just a few examples) and the complexity of these, would have led to an unsustainable increase in connections with a consequent increase in production costs and above all a considerable physical footprint ". [31]
Each frame is uniquely distinguished by an identifier (ID). Data communication is based on the concept of dominant and recessive bits; the node that transmits dominant bits has priority and this considerably reduces contention times, avoiding possible communication delays between the electronic boards connected to the CAN bus. By exploiting a two-wire balanced line as a physical layer, the system is particularly resistant to electromagnetic disturbances. Furthermore, if a node finds itself having systematic hardware errors it is excluded. This type of communication is widely used in the automotive field. A motor vehicle nowadays can have up to 70 electronic control units (ECUs) for various subsystems, from the engine control unit, to the transmission unit, airbag, anti-lock braking / ABS, cruise control, electric power steering, audio, power windows, doors, mirror adjustment, battery and systems for hybrid or electric vehicles and many more. All these control units are connected to each other and are able to communicate with each other thanks to the CAN-bus. Each control unit is also referred to as a CAN node, and all nodes are connected to each other through a two-wire bus with a nominal 120 Ω resistance according to the ISO 11898-2 standard.
Regarding the future implementation of the machine state, I would like to highlight how a microcontroller can manage different data flow in a non-overlapping fashion. The CPU is capable to understand that a peripheral has a data that is ready for being read through two possible methods, the

polling and the interrupt. In the case of polling the software periodically reads the status of each peripheral and when a peripheral need to be serviced the corresponding service routine is executed, the advantage here is that this implementation is very easy, the disadvantage is the high latency and the wasted CPU time. Regarding the interrupt a dedicated line, named IRQ (interrupt request), connects the peripheral with the CPU and in case the peripheral need to be serviced it asserts the IRQ line having low latency but a higher hardware complexity. In case of polling the latency depends on the order in which the peripherals are polled, in case of interrupt, the latency depends on the number of requests simultaneously asserted. In this project we have to deal with a number of different tasks each one performing a different job, so we have to find a way to understand how they interact with each other and which one is more important than the others to be able to define the priority of each task compared to the other ones. Each task is characterized by a state during its execution, so we have the running state (the instructions that compose our task are being executed), waiting and ready state, in both cases the process is waiting to be executed but in ready state the process has already all the needed resources needed for running but the CPU and the terminated state. A scheduling algorithm can be used to put each process in the right state, one has to run, others have to be blocked, others remain in the ready state and a context switch can be defined as the sequence of operations that are performed whenever we have to replace a running process with another one. A priority-based scheduling is a scheduling algorithm in which preemption of the running task happens only if a ready task has a priority greater than that of the running one.

The next steps starting from development carried out in this work are essentially two: the development of a 4G/5G shield for the network communication and the implementation of a state machine for the dialogue and cooperation of all devices in real time and the development of the communication microcontroller-CAN through STM32. Other possible future developments, as I mention at the end of this work, can be the V2V and the V2I technologies.

3.2 Devices, sensors and adopted interface

The choice behind every component is strategic pursuing the goal of having a clear understanding of what's going on around the vehicle, in terms of position, time and other crucial parameters. The adopted devices are highlighted below.

COMPONENT	CONNECTIVITY	PIN
GPS	USART2	PD6(RX) PD5(TX)
BMP180	I2C1	PB8(SCL) PB9(SDA)
CAN	CAN1	PD0(RX) PD1(TX)
OLED DISPLAY	I2C1	PB8(SCL) PB9(SDA)
SD CARD	SPI1	PA5(SCK) PA6(MISO) PA7(MOSI)
RTC	I2C1	PB8(SCL) PB9(SDA)

Table 6: Connections for the different devices

The RTC is a low-cost real-time clock that works with I2C. "The device incorporates a battery input and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line. The RTC maintains seconds, minutes, hours, day, date, month and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator". [32]

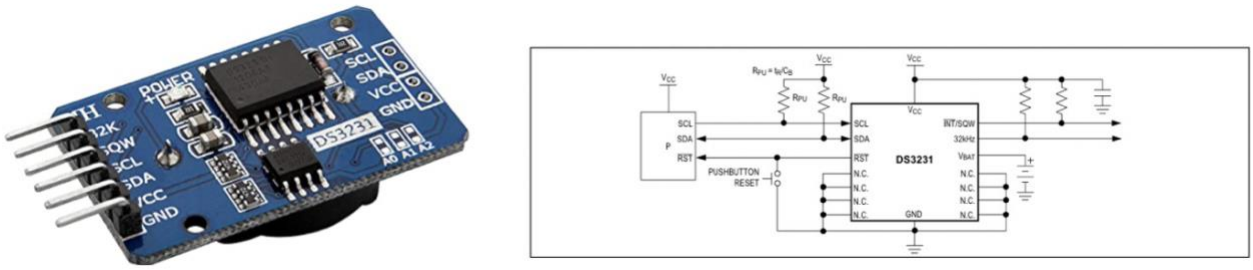

Figure 44: DS3231

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply voltage	Vcc		2.3	3.3	5.5	V
	Vbat		2.3	3.0	5.5	V
Logic 1 input SDA, SCL	Vih		0.7x Vcc		Vcc+0.3	V
Logic 0 input SDA, SCL	Vil		-0.3		0.3xVcc	V

Table 7: DS3231 Datasheet

The OLED display is a 0.96 inches screen that is characterized by a resolution of 128x64 pixel and working voltage range from 3.3V to 5V and it is used with I2C communication protocol. The OLED display is used to visualize on the screen the data coming from the other sensors, like temperature,

altitude, pressure, hour, date, GPS data. This device is really important especially in the debug environment to understand if a sensor behaves as expected or not, in a clear and easy way, by visualizing the output on the screen, displaying the results of computation to a final user.

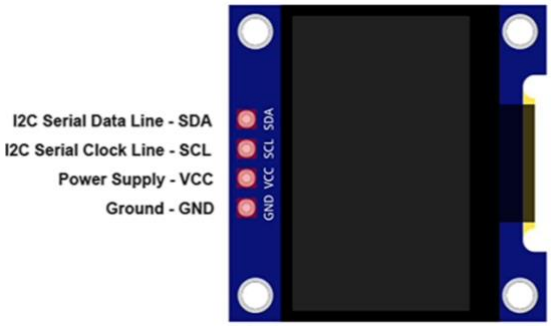

Figure 45: SSD1306

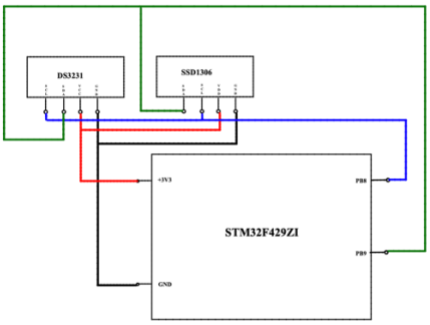

Figure 46: SSD1306 Connection

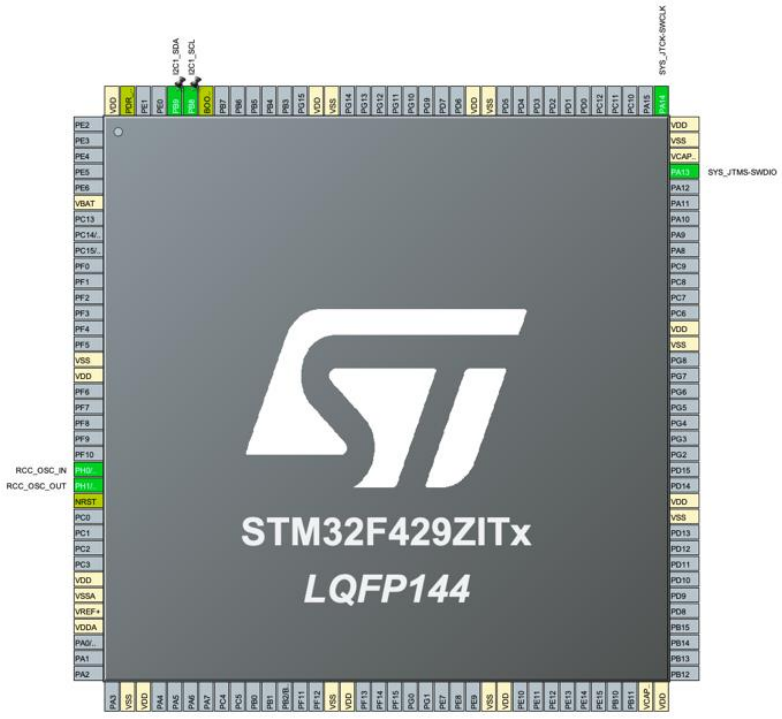

Figure 47: SSD1306 Pinout

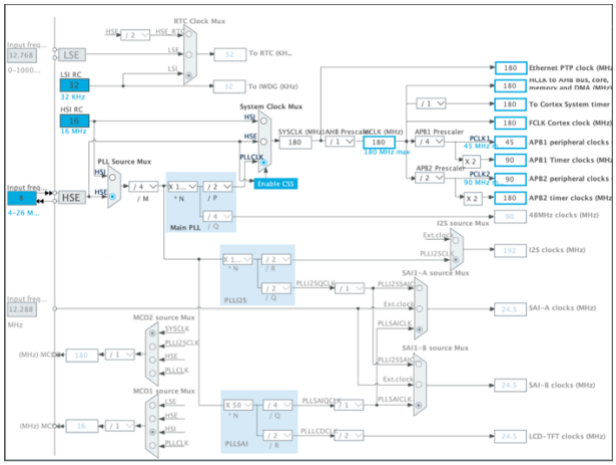

Figure 48: SSD1306 Clock Configuration

Figure 49: SSD1306 Display

The BMP18O is a sensor able to acquire temperature, pressure and altitude. It supports I2C communication protocol and it is intended to be connected to a microcontroller. “The BMP180 consists of a piezo-resistive sensor, an analog to digital converter and a control unit with EEPROM and a serial I2C interface. The BMP180 delivers the uncompensated value of pressure and temperature and these data have to be compensated by the calibration data of the EEPROM of the BMP180”. [33]

It is essential to understand the environment in which the vehicle is moving, the weather, the pressure at which the vehicle is subjected and the altitude reached by the vehicle, so there are a lot of information that can be deduced by these parameters.

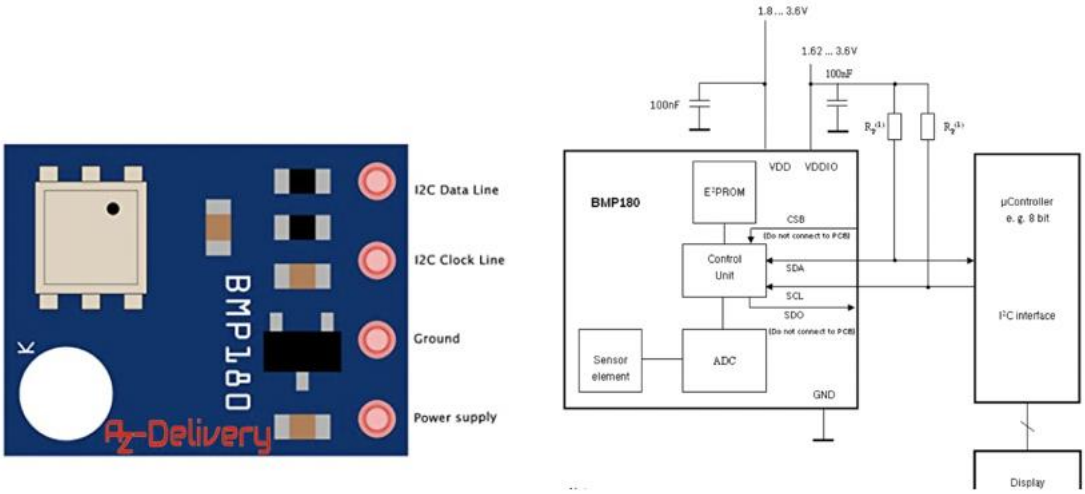

Figure 50: BMP180

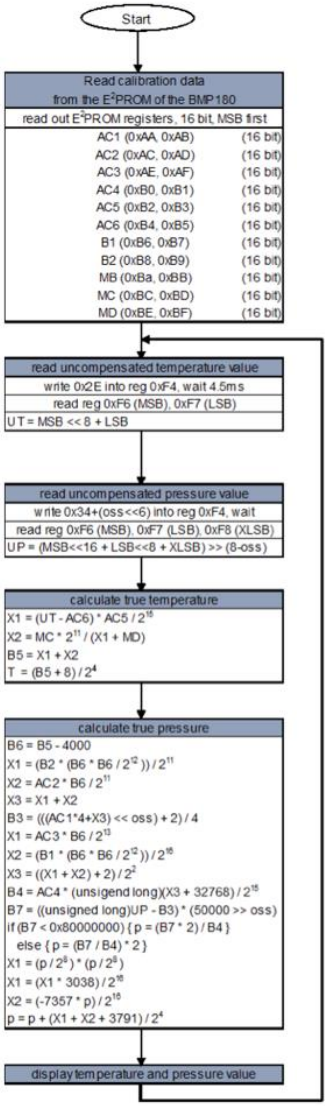

Figure 51: Algorithm for pressure and temperature measurement

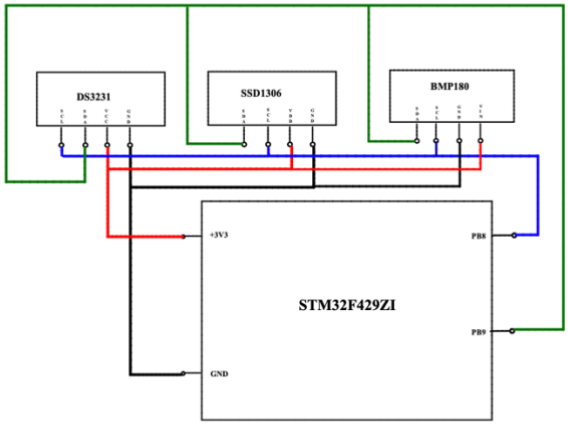

Figure 52: BMP180 Connection

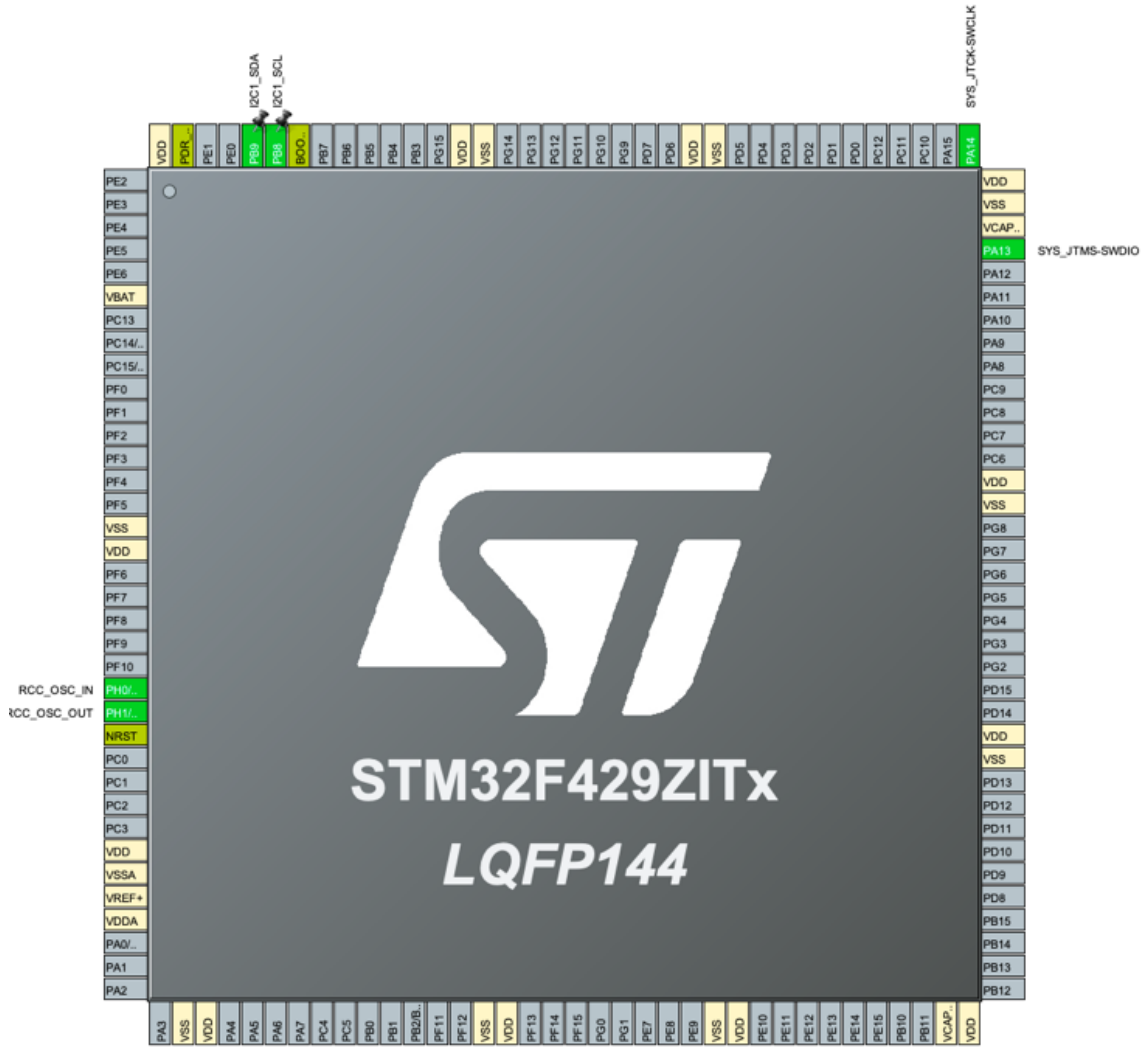

Figure 53: BMP180 Pinout

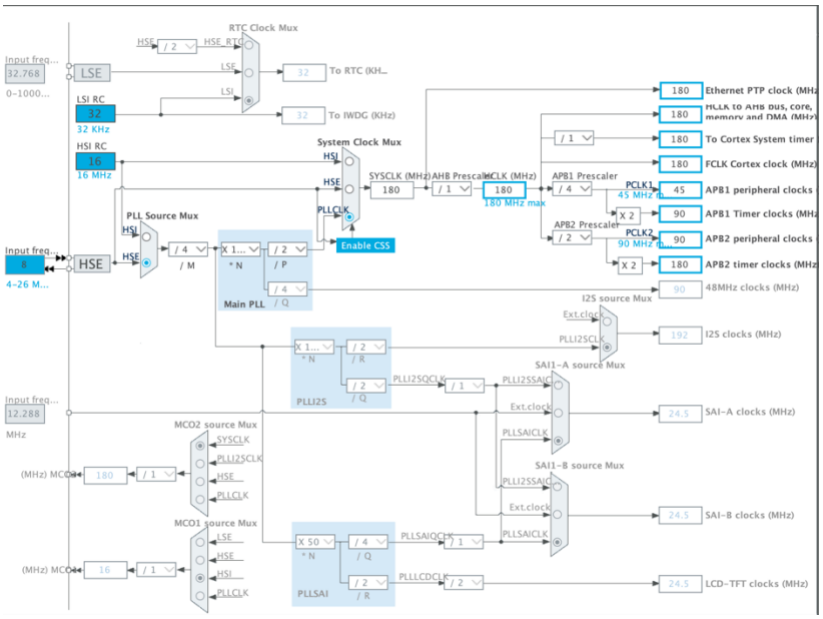

Figure 54: BMP180 Clock Configuration

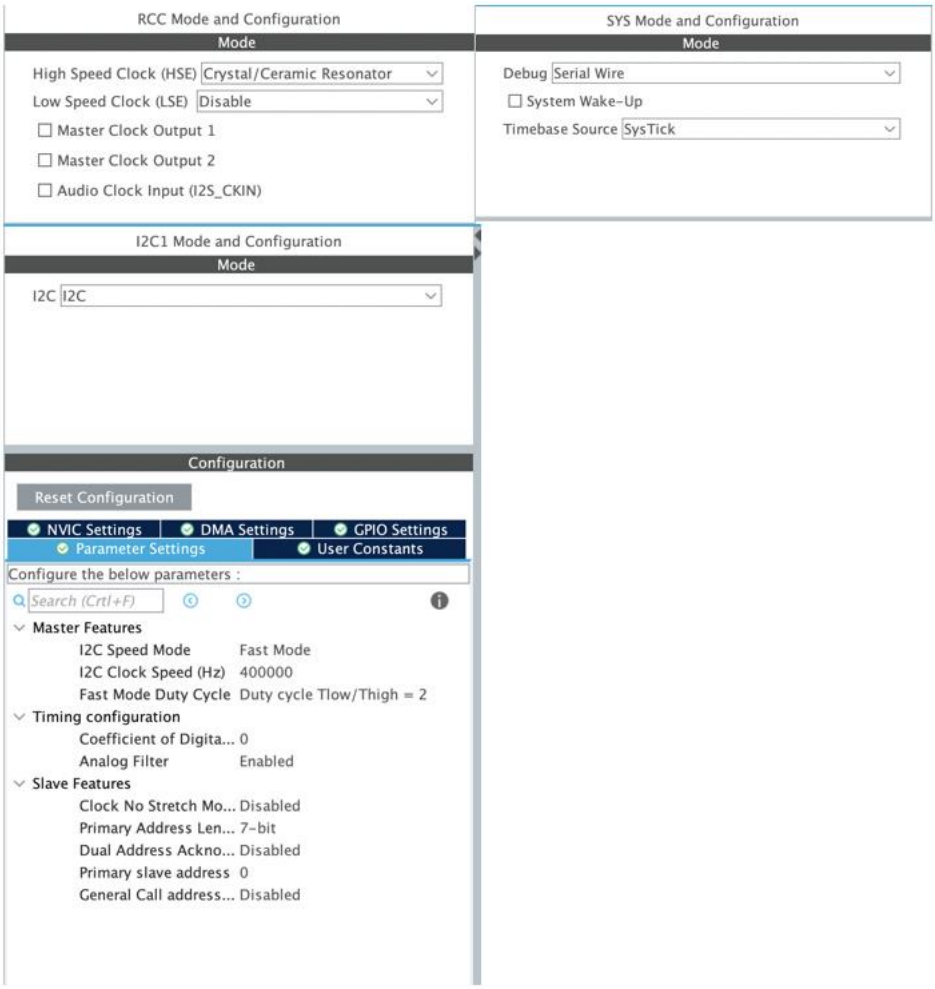

Figure 55: BMP180 RCC, Sys, I2C Configuration

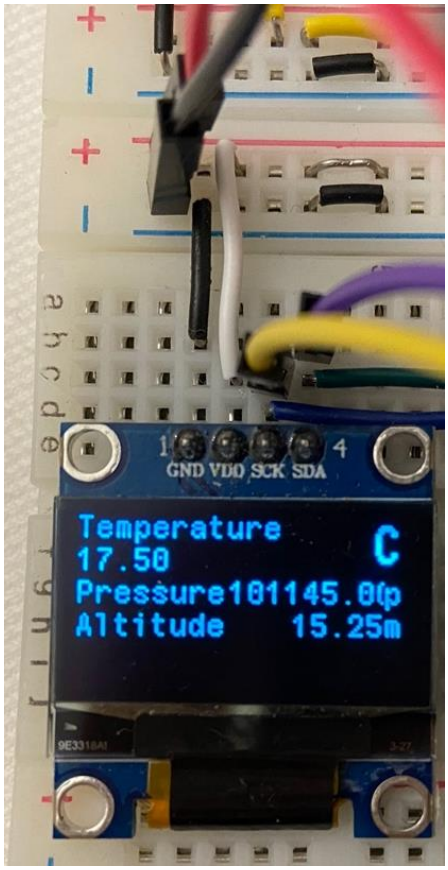

Figure 56: BMP180 Results

GPS is needed to keep track of the vehicle's position, so it is crucial to be able to obverse the vehicle itinerary during the time and to let the other vehicles, users, infrastructures to know where the vehicle is located in real-time, allowing vehicles cooperation and, for example, rescue operations. It makes use of one of the NEO-6 (in particular NEO-6M-0-001) module series that is "a family of stand-alone GPS receiver featuring the high-performance u-blox 6 positioning engine. Their compact architecture and power and memory options make NEO-6 modules ideal for battery operated mobile devices with very strict cost and space constraints." [34] The NEO6 GPS interacts through USART with micro, once received all NMEA messages, I wanted to obtain only the parameters referred to the GPRMC, to do so, after receiving all the messages I separated them by the \n (as can be seen in Figure 61) to obtain single NMEA messages and then through an if-logic I selected only the GPRMC messages (there was also a check about the checksum) to enter in a switch case to obtain all the needed values (utc time, longitude,latitude, north-south, east-west, validity and so on) shown in Figure 63 and Figure 64.

Figure 57: GPS NEO6M

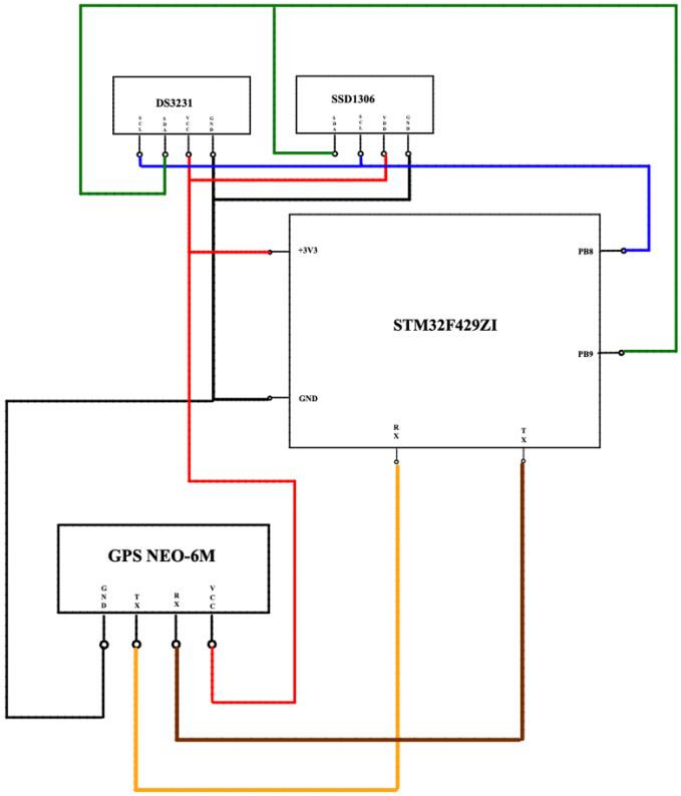

Figure 58: GPS NEO6M Connection

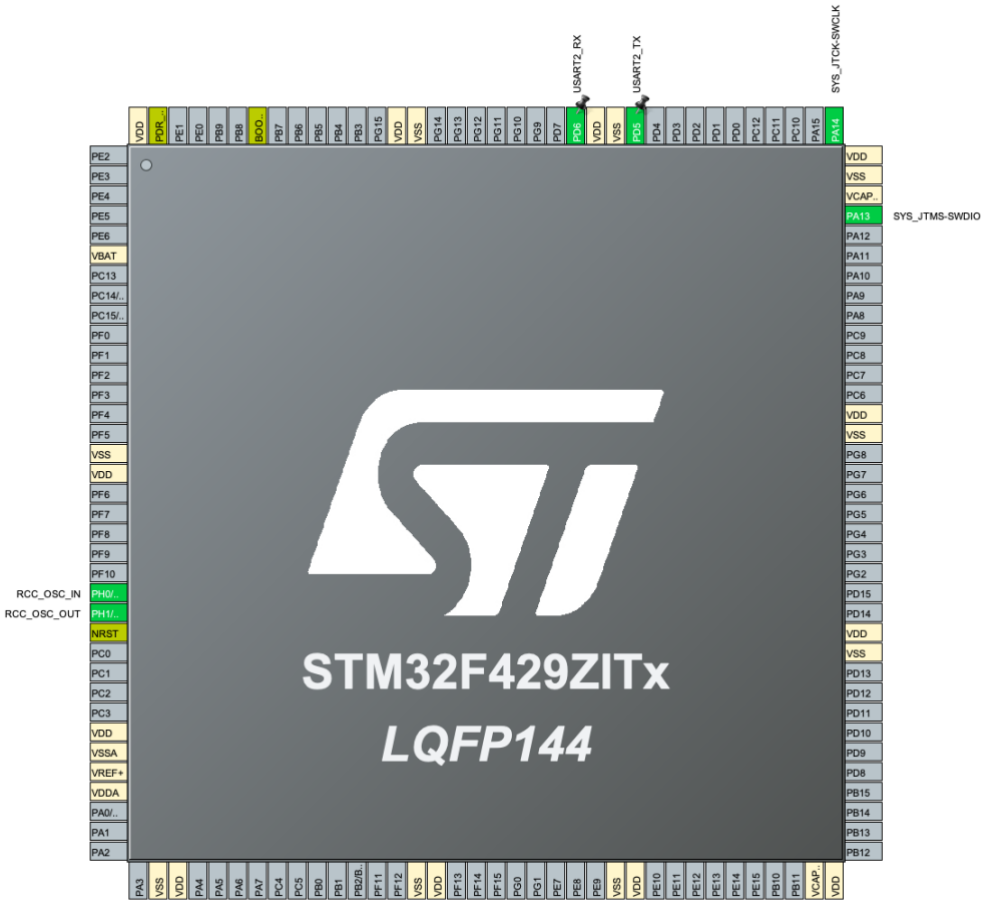

Figure 59: GPS NEO6M Pinout

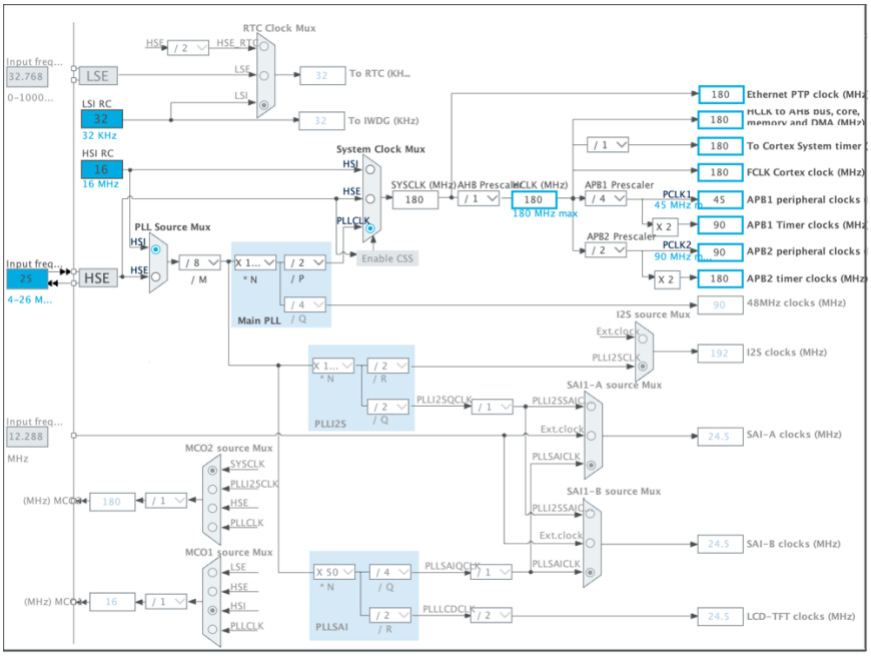

Figure 60: GPS NEO6M Clock Configuration

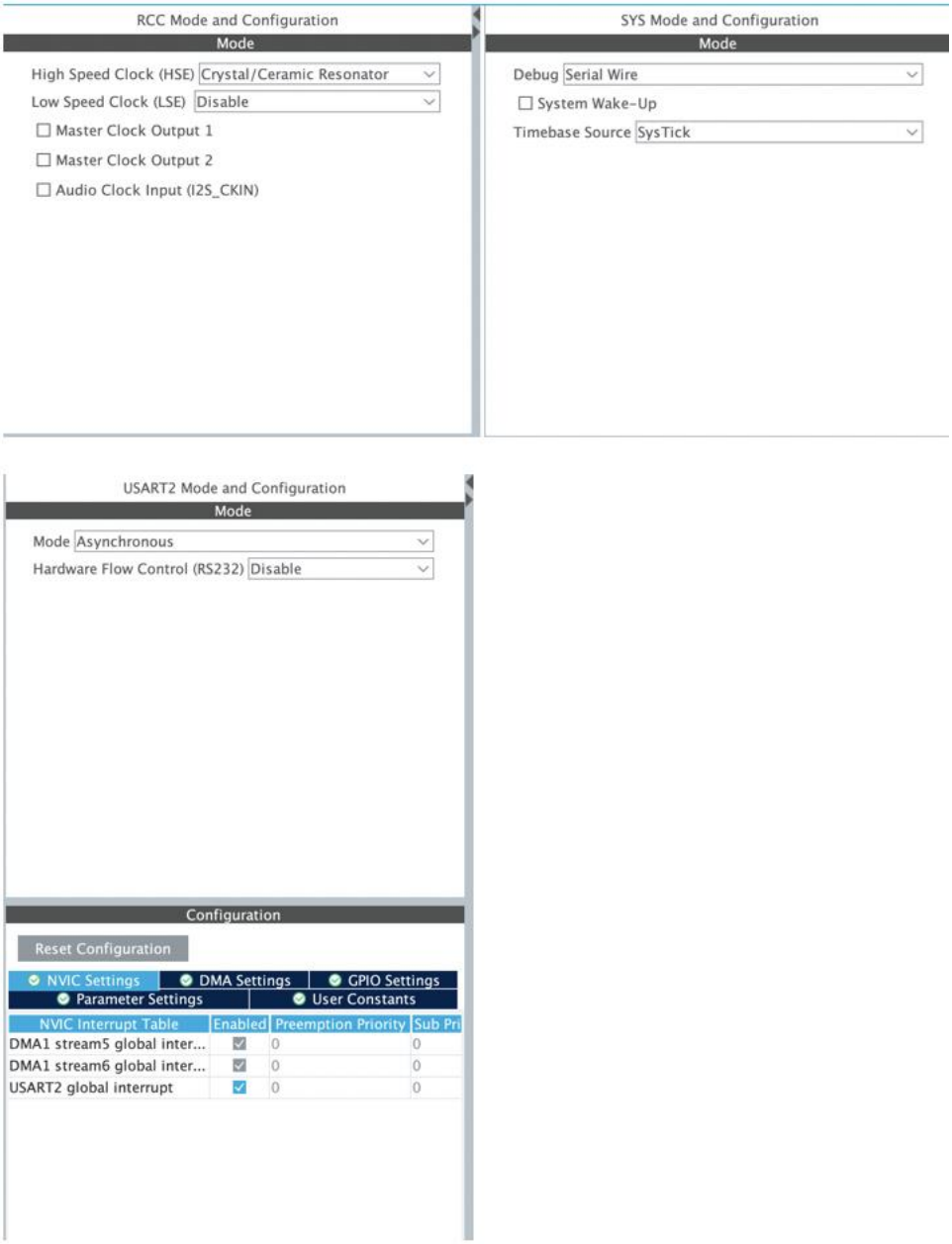

Figure 61: GPS NEO6M RCC, Sys, USART Configuration

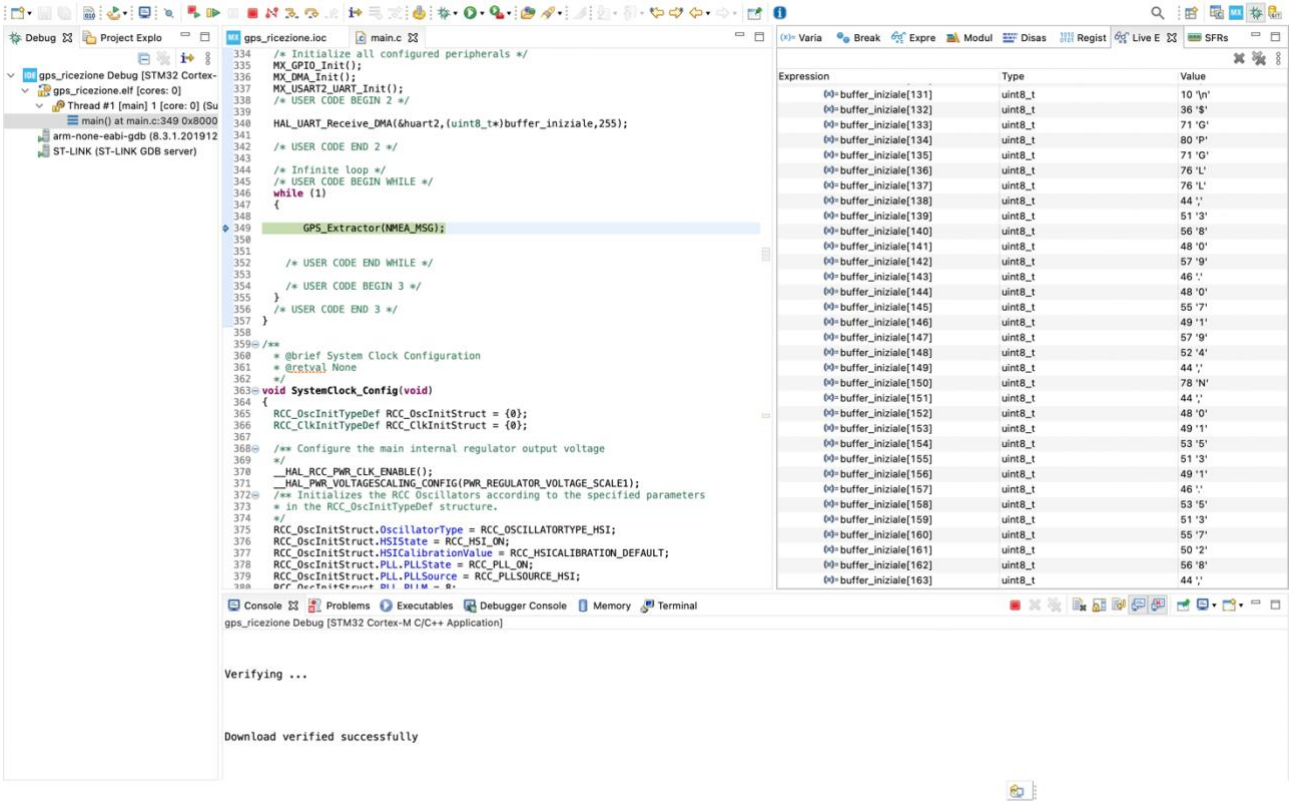

Figure 62: GPS Buffer iniziale

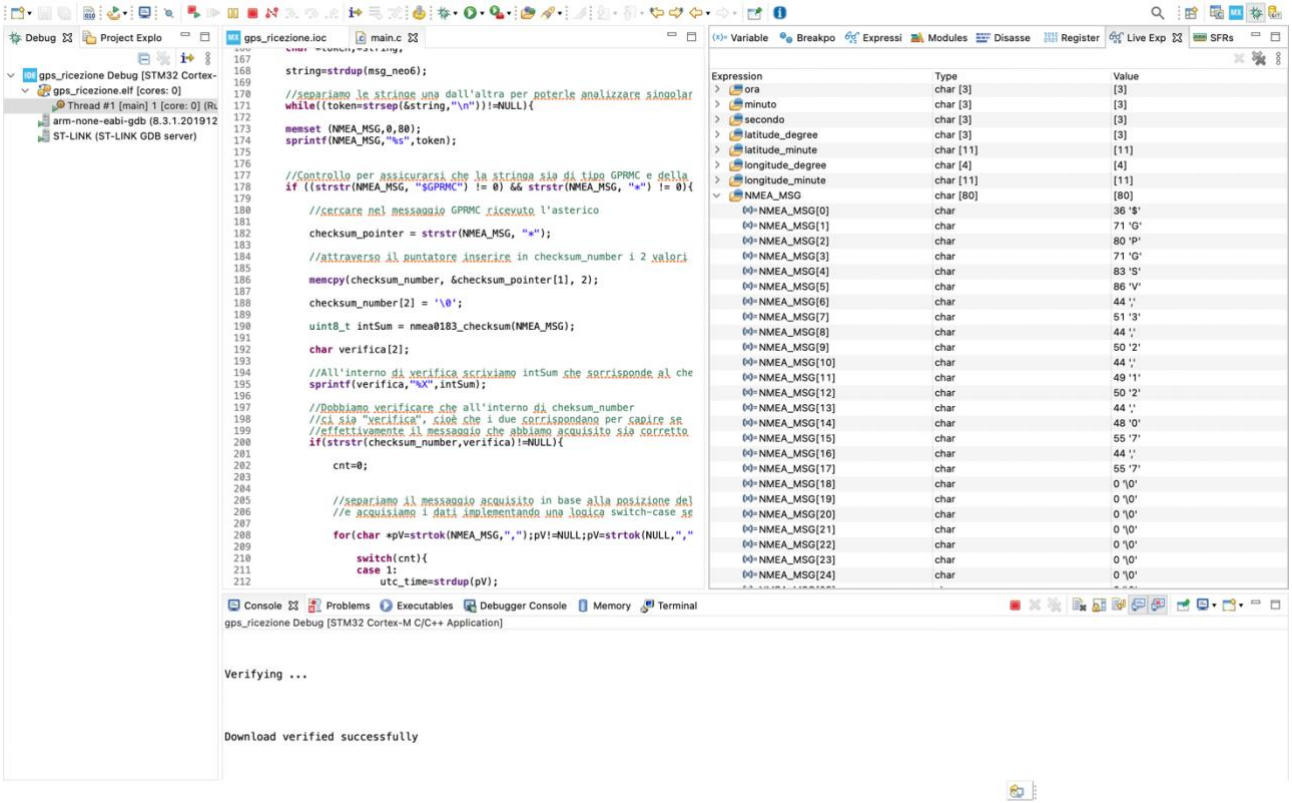

Figure 63: GPS NMEA_MSG

checksum_number	char [3]	[3]
checksum_number[0]	char	55 '7'
checksum_number[1]	char	52 '4'
checksum_number[2]	char	0 '\000'
validity	char *	0x2002fa60 "A"
Add new expression		

Figure 64: GPS Checksum and Validity

nmea_latitude	char *	0x20016078 "3809.07478"
latitude_degree	char [3]	[3]
latitude_degree[0]	char	51 '3'
latitude_degree[1]	char	56 '8'
latitude_degree[2]	char	0 '\000'
latitude_minute	char [11]	[11]
latitude_minute[0]	char	48 '0'
latitude_minute[1]	char	57 '9'
latitude_minute[2]	char	46 '.'
latitude_minute[3]	char	48 '0'
latitude_minute[4]	char	55 '7'
latitude_minute[5]	char	52 '4'
latitude_minute[6]	char	55 '7'
latitude_minute[7]	char	56 '8'

Figure 65: GPS latitude

nmea_longitude	char *	0x20016098 "01531.53707"
longitude_degree	char [4]	[4]
longitude_degree[0]	char	48 '0'
longitude_degree[1]	char	49 '1'
longitude_degree[2]	char	53 '5'
longitude_degree[3]	char	0 '\000'
longitude_minute	char [11]	[11]
longitude_minute[0]	char	51 '3'
longitude_minute[1]	char	49 '1'
longitude_minute[2]	char	46 '.'
longitude_minute[3]	char	53 '5'
longitude_minute[4]	char	51 '3'
longitude_minute[5]	char	55 '7'
longitude_minute[6]	char	48 '0'
longitude_minute[7]	char	55 '7'

Figure 66: GPS longitude

In order to store all the data previously acquired and to save them (in a machine state scenario in which it is implemented a saving state after a configuration, read and acquisition one) we have to use a system able to perform this task, so we used a Micro SD card communicating with our microcontroller through SPI "for reading and writing through the file system and the SPI interface driver" [35] data coming from the sensors. VCC is the power supply and it is in range between 4.5V and 5.5V, MISO, MOSI, SCK and CS (chip select) are for the SPI. For the SD card I have interfaced this device with the BMP180 sensor, first I created a file inside the SD Card Directory, at that point I acquired temperature, pressure and altitude through the BMP180 sensor and I inserted with a specific function the values previously collected at the step before inside the file created at the step one as can be seen in Figure 70.

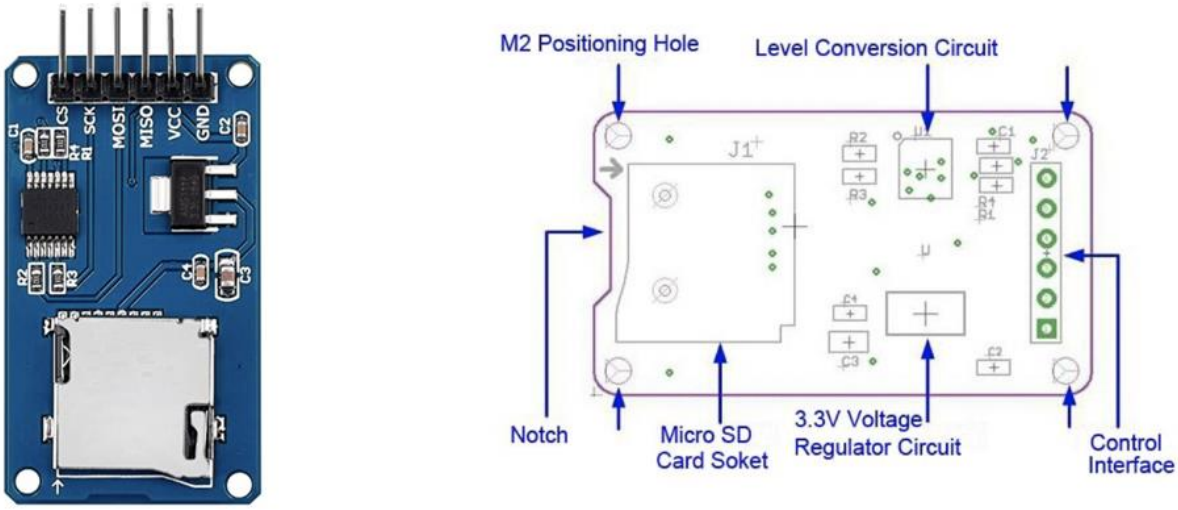

Figure 67: SD CARD

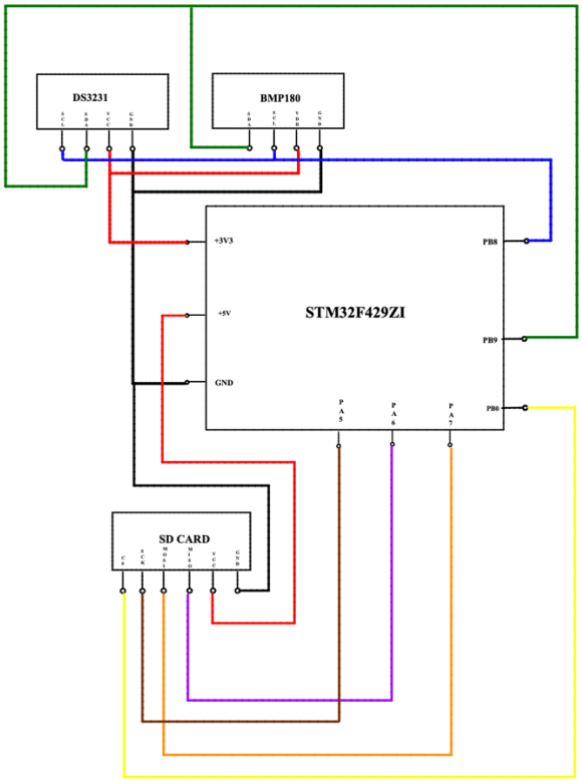

Figure 68: SD CARD Connection

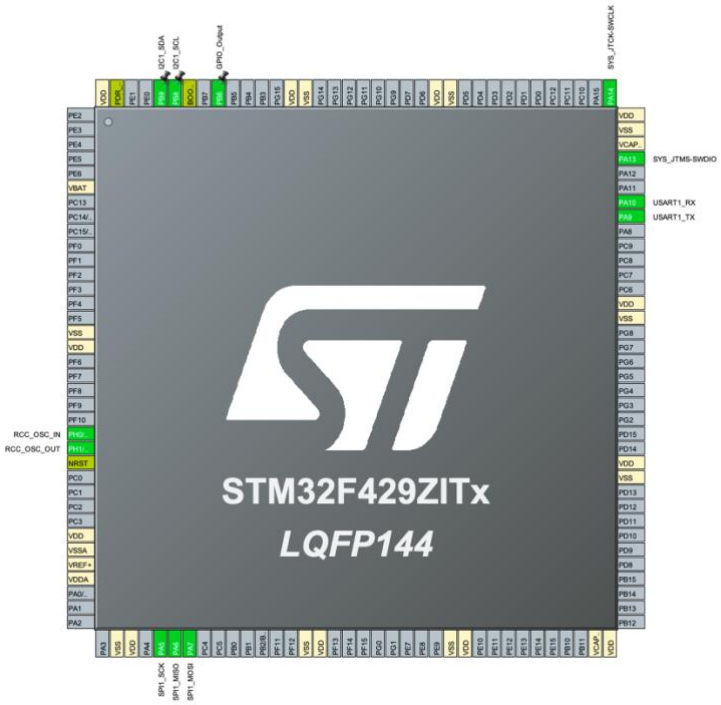

Figure 69: SD CARD Pinout

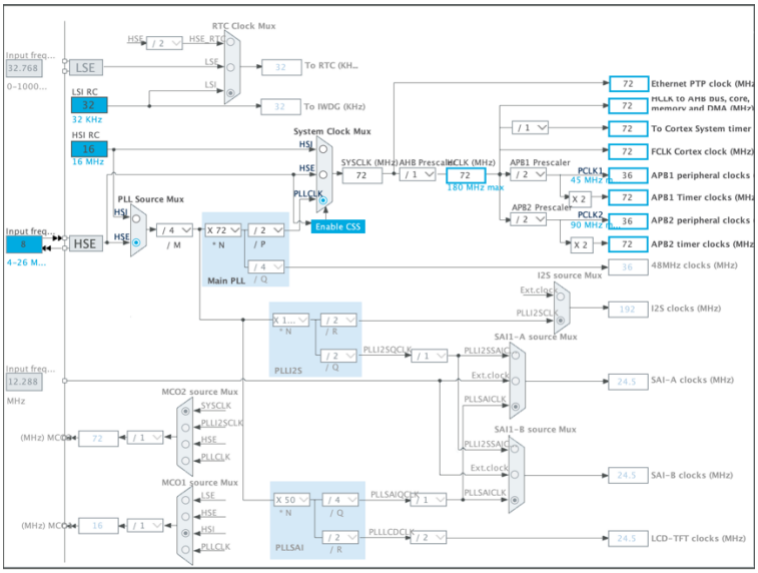

Figure 70: SD CARD Clock Configuration

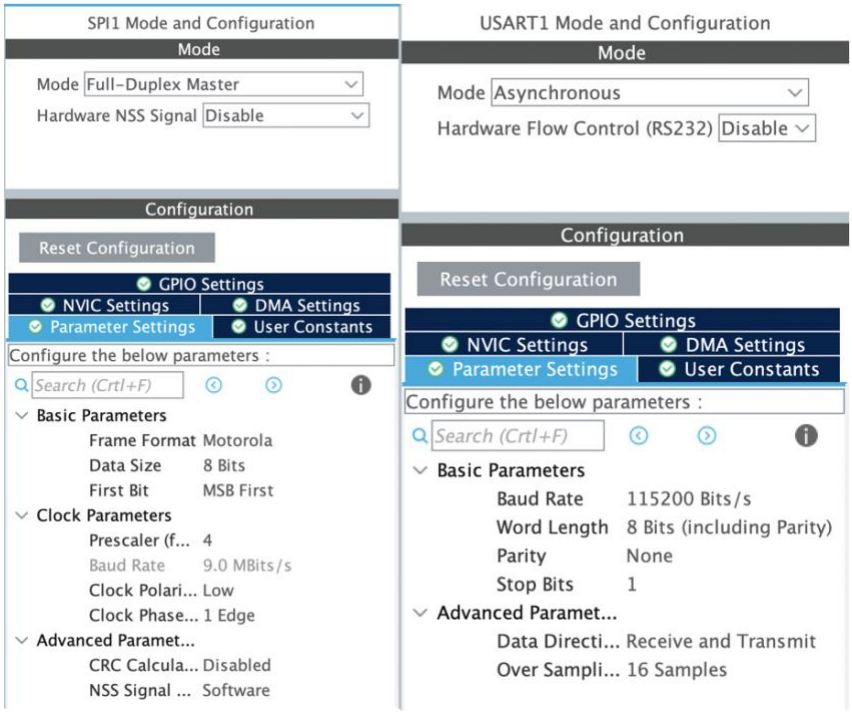

Figure 71: SD CARD SPI and USART Configuration

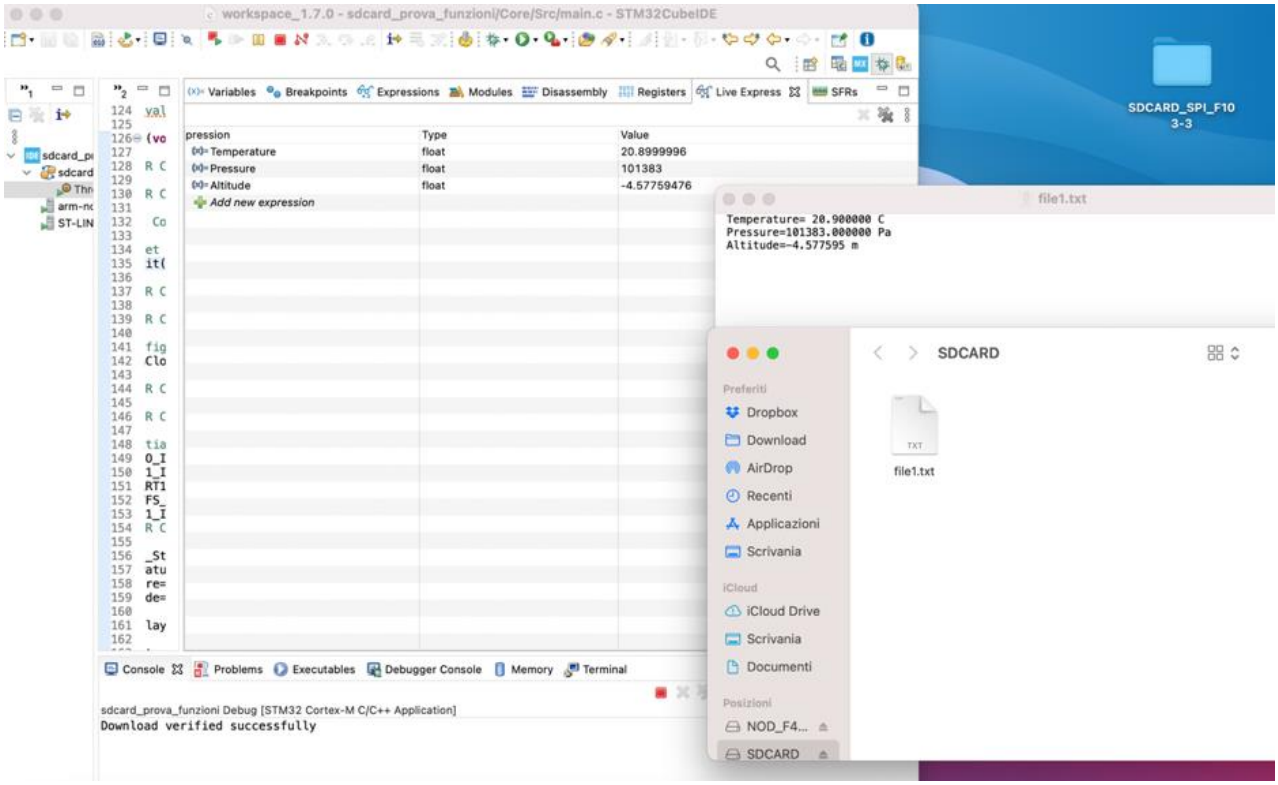

Figure 72: SD CARD Results

Chapter 4: Road Test and future scenarios

4.1 Test on the vehicle

Given that the timing foreseen for the development phase of the CAN transceiver and GSM shield were not compatible with the drafting of the thesis, to speed up the representation of the possibilities offered by the described platform, an analogous interface based on the ARDUINO system has been developed in collaboration with CNR ITAE. While the development scenario imagined on the STM32 microcontroller remains unchanged, the tests carried out on the vehicle are described below. With the support of CNR ITAE, I used an Arduino to be able to acquire the needed variables to understand the propulsive system of the vehicle. The system is able to acquire environmental data, vehicle data from CAN network, position and it is able to send data to a server for later observations. This system requires a lot of different shields.

The test consists in showing the data acquired by the ITAE server during a test drive of the hybrid vehicle. The purpose of this test drive was to underline the operating activation of the fuel cell and how the fuel cell can influence the other observed parameters. The path of the test drive is shown in figure 73, the starting condition of the state of charge of the battery was 62% and the selected threshold to control the activation logic of the fuel cell has been set as 60-70, this means that below 60% the fuel cell switches on and above 70% it switches off (see Figure 26, Chapter 2). The yellow line indicated the return trip, once the test drive was over, we left the vehicle stationary and turned on to observe the battery recharging through the fuel cell, so we saw all the needed steps to clarify the behavior of the fuel cell system: the activation, the battery recharging condition and the turn off condition.

On February 21, 2022, a field test was carried out on board the hydrogen-electric hybrid minibus to acquire the variables of interest via server, to verify the acquisition of the same variables through the CAN bus on board the vehicle, for an analysis of the obtained results and to highlight the activation and functioning logic of the fuel cell and to show the interaction of the various systems of interest, i.e. fuel cell, battery pack and electric motor.

The variables of interest for these 3 systems are the following: Motor speed and motor power as regards the electric motor, current, voltage, power and pressure of the hydrogen tanks for the fuel cell and finally current, voltage, state of charge and power for the battery pack. The route is divided into two parts, round trip, between the two phases the vehicle was turned off to observe the dynamics of the systems during shutdown and finally the vehicle, brought back to the i-Next hydrogen production center from where it started, it remained on but stopped to allow you to observe the charging of the battery via fuel cell. As already shown in the previous chapters, the fuel cell activation logic depends on the state of charge of the batteries, at the beginning this value was 62% as shown in figure 74, and the set thresholds were 60-70, i.e. the fuel cell had to turn on as soon as the state of charge of the battery dropped to 60% and turn off when the same parameter was greater than 70%. Furthermore, once activated, the fuel cell works at constant power, this parameter can be modified and has been set for the described test equal to 10kW; if the electric motor requires less power than that fuel cell can supply, the fuel cell recharges the battery pack, otherwise it helps the battery pack to move the vehicle (see Figure 25, Chapter 2).

Once described the scenario in which the test took place, we can see in figure 75, 76, 77, 78 the graphs related to the way there. In figure 74, in addition to the starting and switching off points of the vehicle as in figure 78, the points where the battery recharges have been highlighted, in fact, peaks related to the engine speed below zero can be noted, this is because in those points the motor reverses its direction of rotation and therefore an inversion of the energy flow in the battery is obtained, resulting in input power to the battery and highlighting the effect of regenerative braking, the same

phenomenon can be seen from figure 78 in the points in which the current and power of the battery pack drop below zero. In figure 77 it is possible to observe the operation of the fuel cell which turns on as soon as the state of charge of the vehicle reaches the minimum threshold set, i.e. 60, as seen in figure 76. After a first start up phase, the current and voltage remain constant over time precisely because the fuel cell works at constant power always delivering 10kW, it can be seen how the pressure of the hydrogen tanks begins to decrease right after the activation of the fuel cell. In the interval in which the fuel cell is active, it is possible to notice how the battery discharge occurs slower than before the fuel cell activation due to the presence of the fuel cell which helps the battery to deliver the power required by the engine.
Graphs 79, 80, 81 refer to the return path, it can be seen from figure 79 that once the vehicle returned to the starting point (i-Next hydrogen production site) it was left on but stopped to observe the operation of the fuel cell as a source of energy for the battery, since the vehicle was stationary, in fact, while remaining on, it had a demand for power that was always lower than that the fuel cell supplied. Figure 80 shows the trend of the variables related to the fuel cell which always works at constant power, including the quantity of hydrogen in the tanks observable through the data about the pressure of the tanks, which at the beginning is almost 80 bar while at the end it is below 50 bar. Finally, it can be clearly observed in figure 81 what was previously described, i.e. the stationary vehicle has a power demand such that the fuel cell, by using the hydrogen in the tanks, supplies energy to the battery, in particular an increase of about 50% to almost 70% of the state of charge can be seen.
A system capable of interacting with other components and transmitting data offers several possibilities and this is the reason why a system such as the one conceived and described in this thesis can be very useful. All the data that a microcontroller is capable of acquiring and managing, in fact, can be used for various purposes, for example predictive maintenance through a real-time update of the main system variables with indicators that detect the possibility of an imminent failure; other applications may concern the remote control of energy management logics that can be implemented remotely or the remote reconfiguration of a system after a careful analysis of the efficiencies performed by expert operators; indication of the refill / recharge points according to the punctual consumption of the vehicle and finally more complex scenarios in which the parameters connected to the vehicle's energy flows (fuel cell activation thresholds or setting of the power delivered by the fuel cell) can interface with the external context (traffic flows, external vehicles) to govern a wider mobility scenario by exploiting the potential of big data.

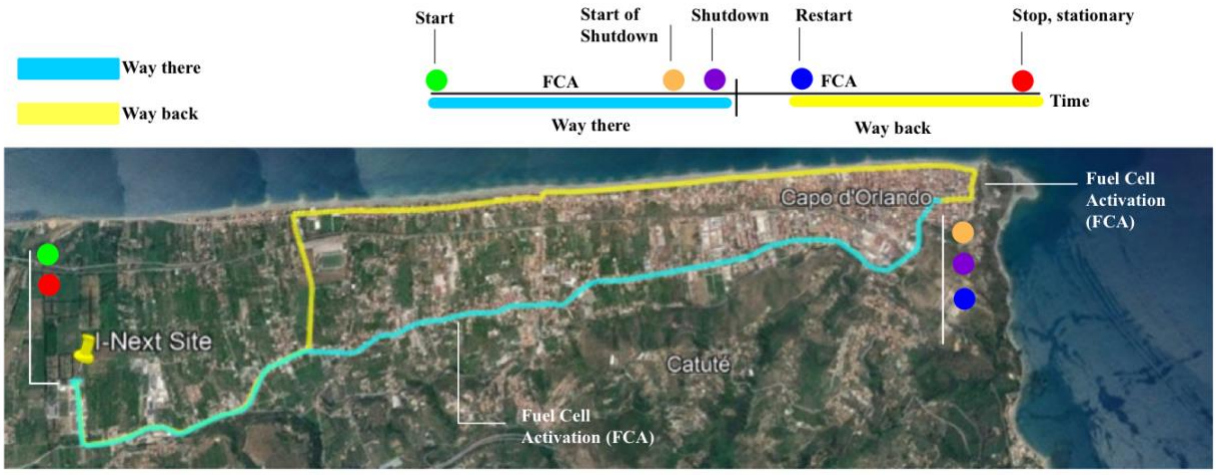

Figure 73: Test drive path

Figure 74: SOC at the beginning of the test drive

Figure 75: Electric Motor parameters on the way there

Figure 76: SOC for fuel cell activation

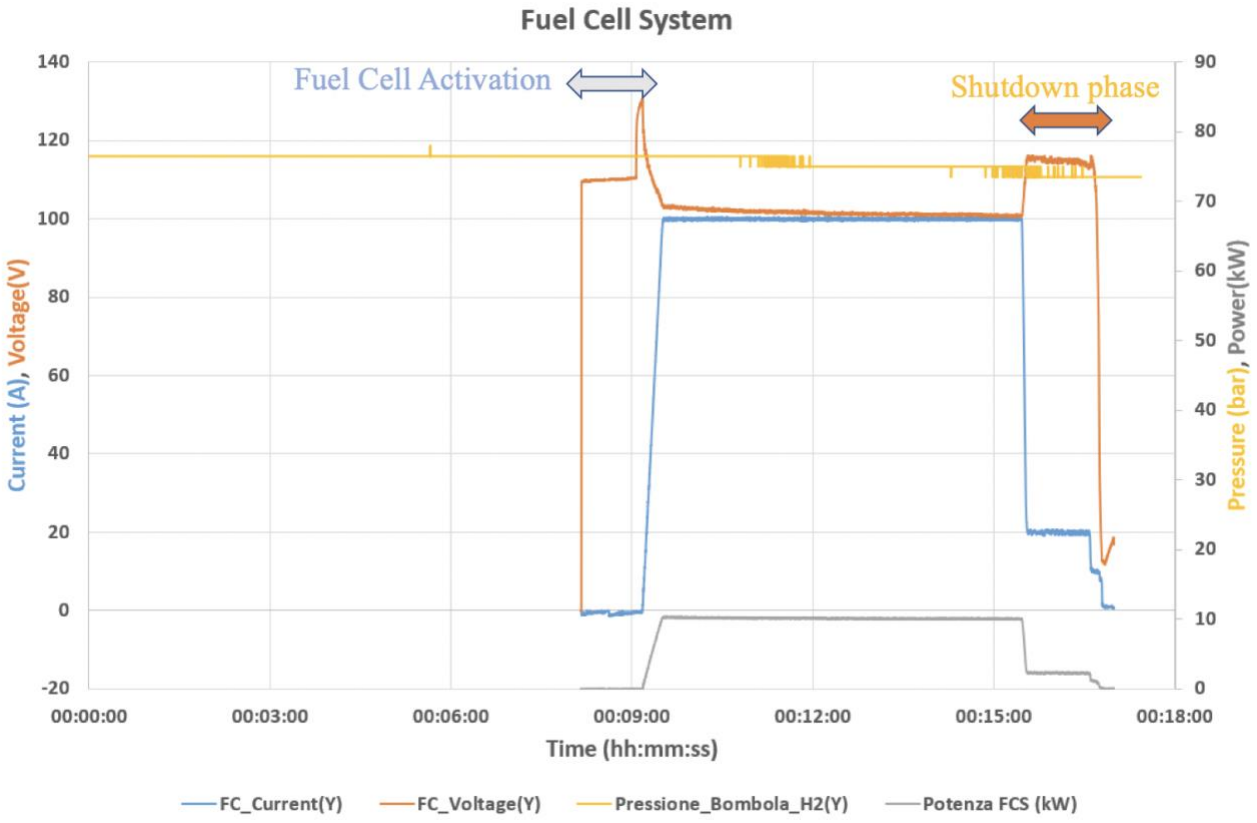

Figure 77: Fuel Cell Systems parameters on the way there

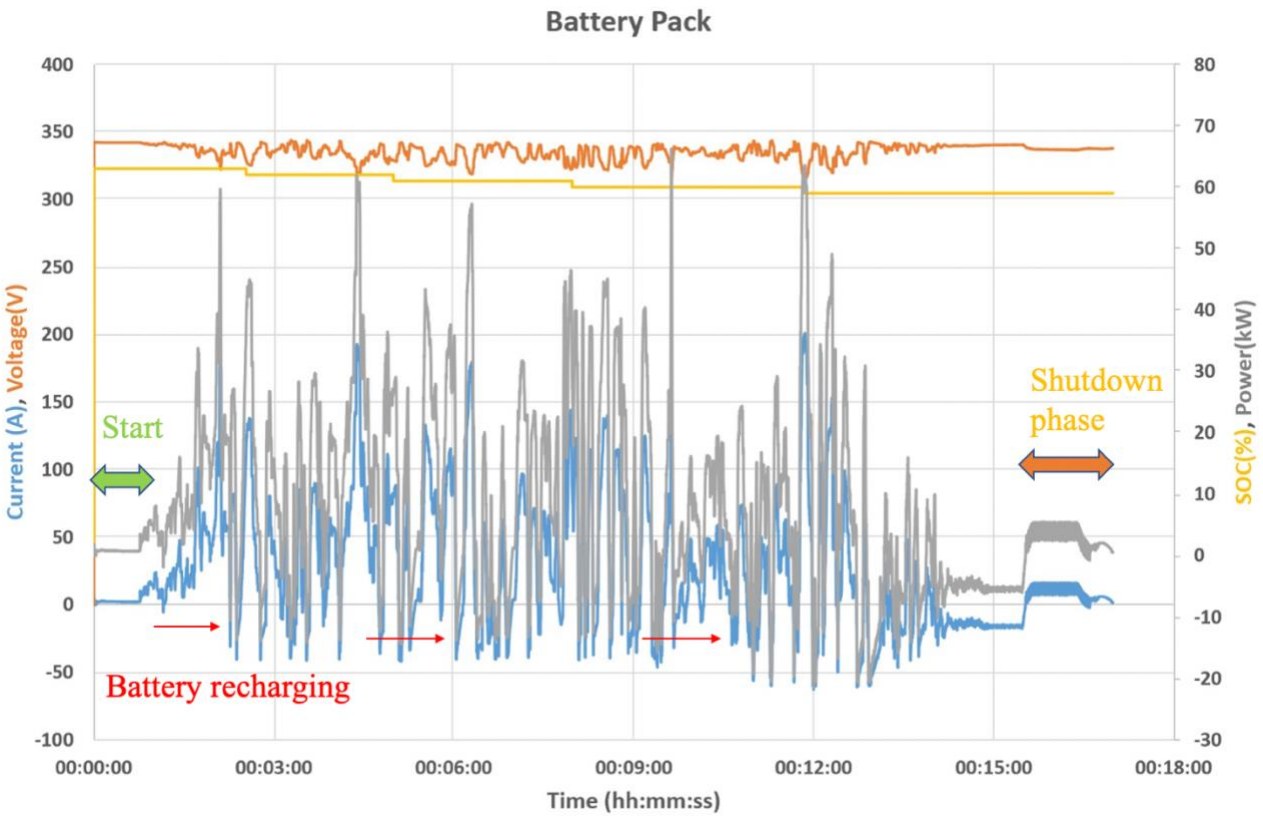

Figure 78: Battery pack parameters on the way there

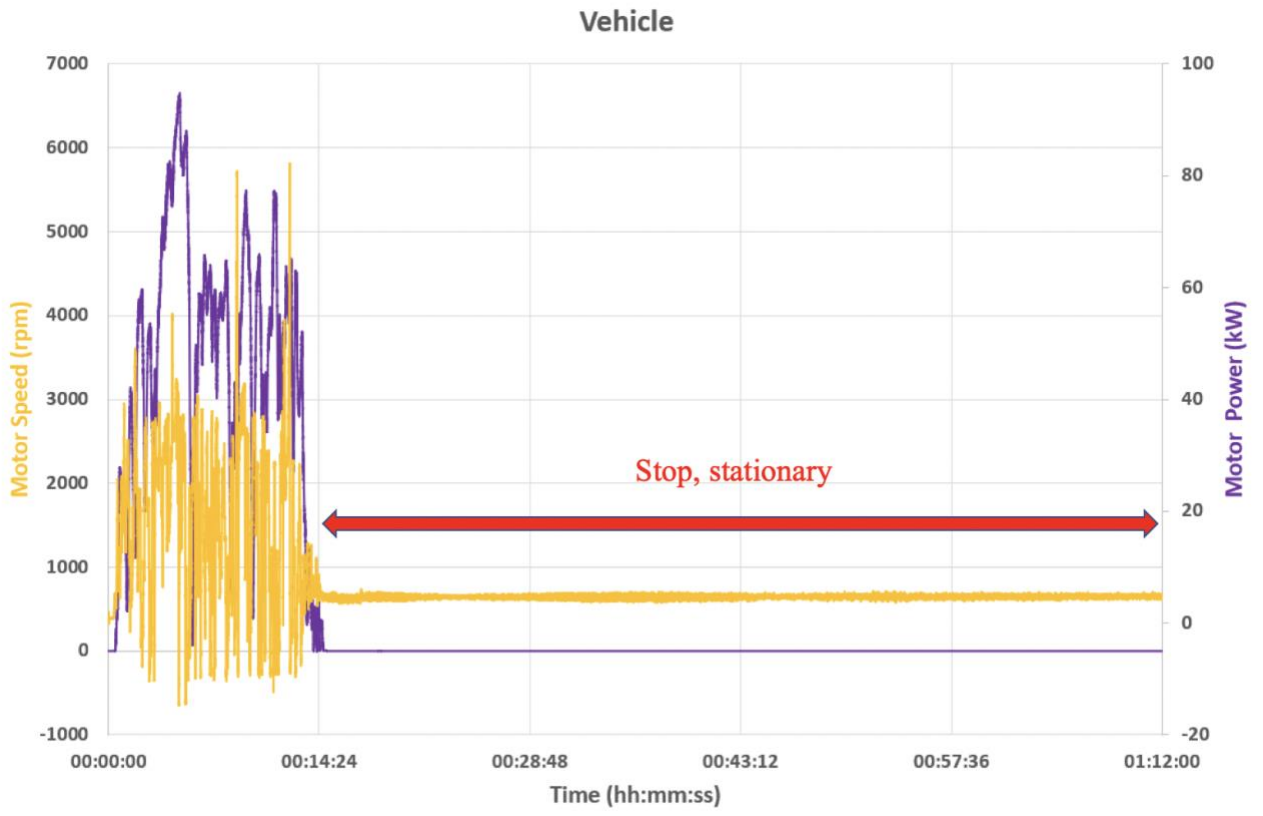

Figure 79: Electric Motor parameters on the way back

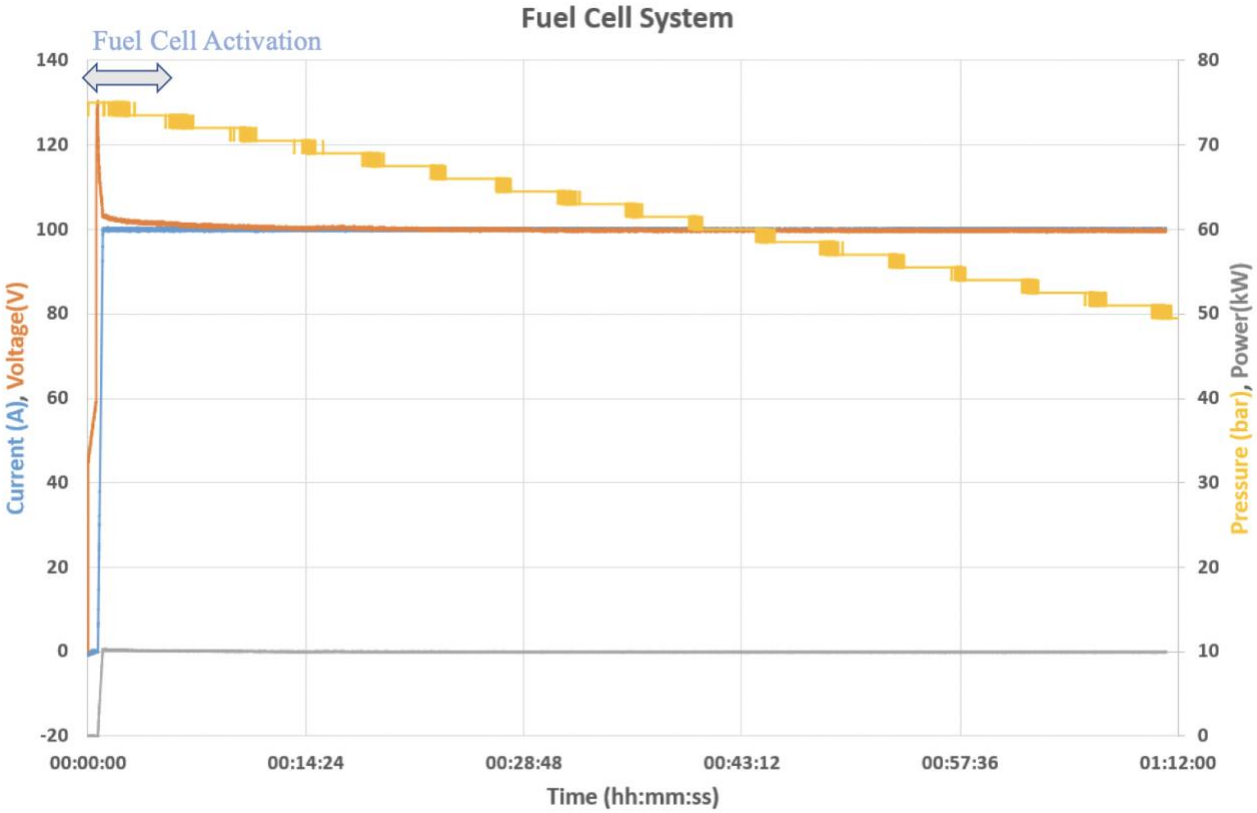

Figure 80: Fuel Cell System parameters on the way back

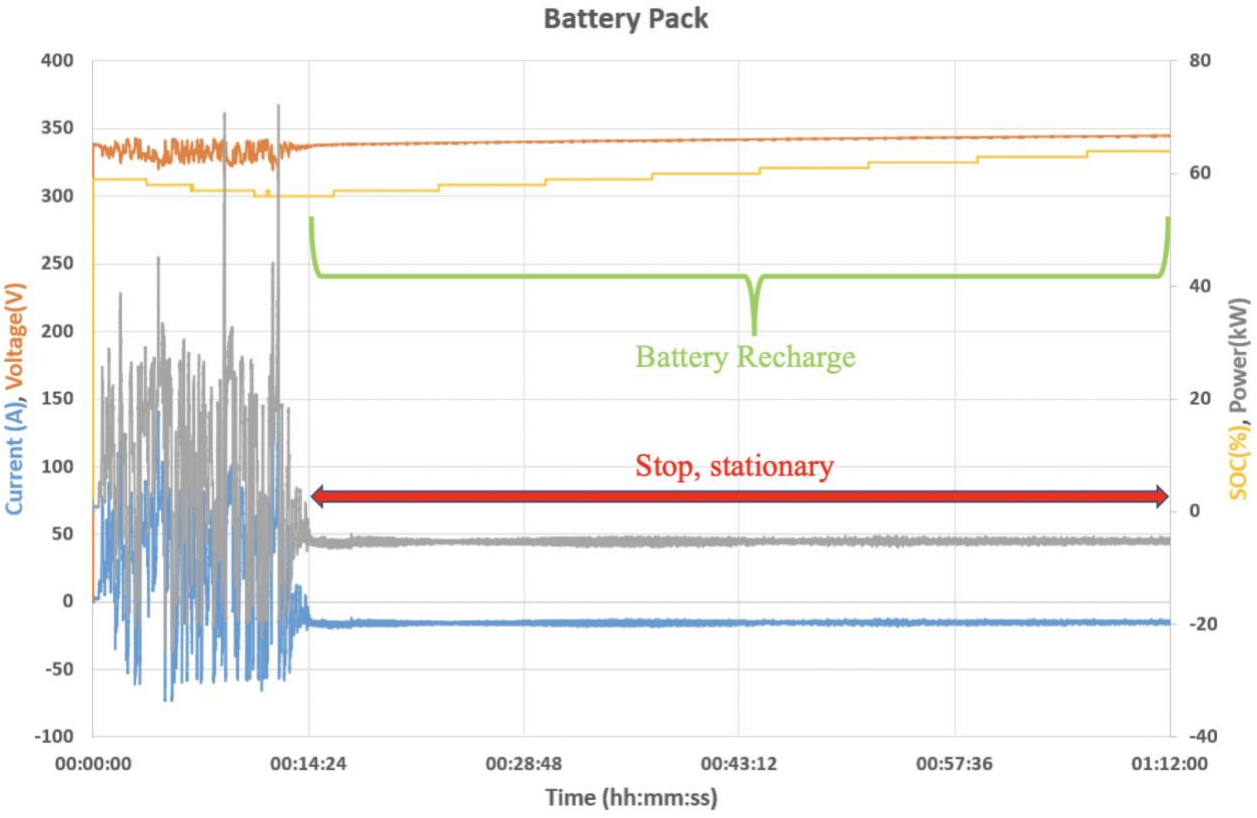

Figure 81: Battery Pack parameters on the way back

4.2 Possible future scenarios

Leaders from Academia, Industry and Government are working together to accelerate the deployment of advanced ITS to achieve the global challenge to increase the road safety. V2X is a typical ITS for safety. The US, Europe and Japan all have the objective to reduce the traffic accidents with the goal of no death or serious injuries. According to the US Department of transportation the main requirements of V2X are that exchanges are non-networked; protocols are interoperable, co-existence in same channel and must be backward compatible. When we say vehicle, we mean any motorized vehicle including cars, buses, trucks, ambulances and motorbike similarly when we say pedestrian, in general terms, we refer to vulnerable road users that include pedestrians, cyclist, scooter rider, disabled; with the infrastructure we mean smart traffic light, road signs and V2X roadside units, finally mobile network can be 4G or 5G.

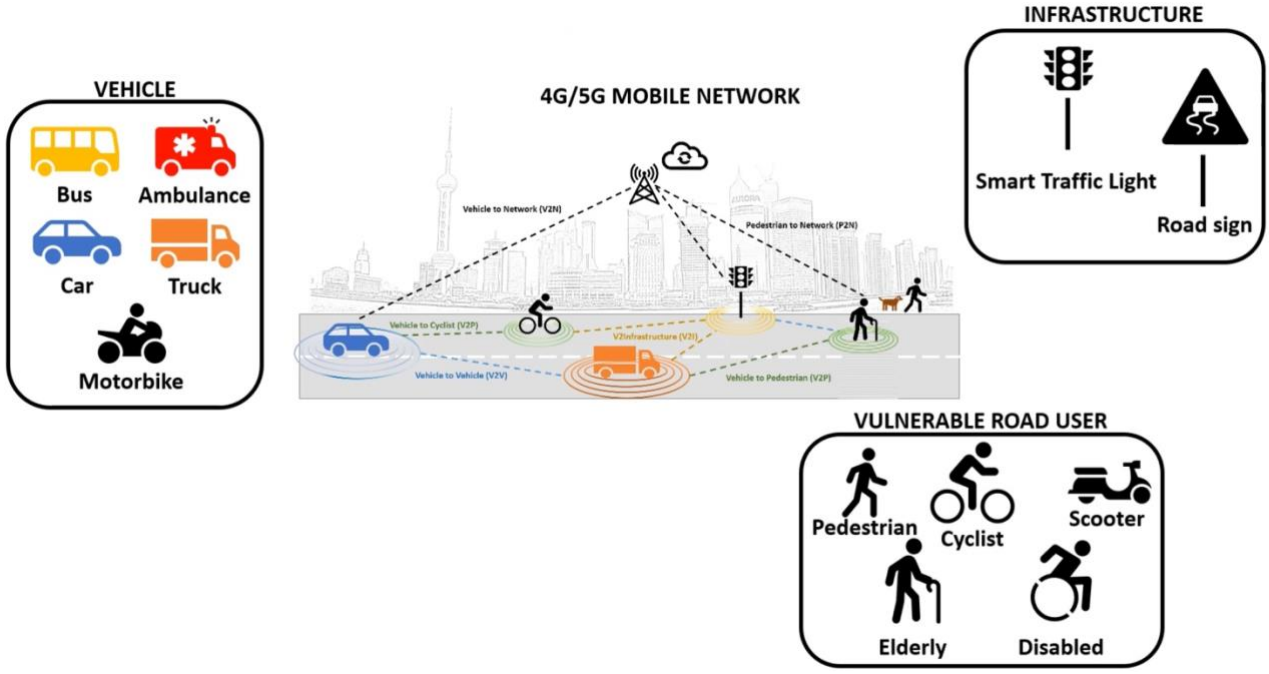

Figure 82: Communication between network, infrastucture, vehicle and users

The use of collective perception message allows vehicles to see what they otherwise could not, further distances and around obstructions such as curves and buildings. Connected infrastructure can spot a non-connected vehicle and non-connected vulnerable road users, such as pedestrian, bicycles, scooters and motorcycles. That information is transmitted to other vehicles whose drivers do not have a line of sight to the non-connected objects allowing the vehicles to react ahead of time faster than they could otherwise, this advance warning can reduce accident frequency and potentially save lives even when not all of the vehicles involved are connected. Vehicle-to-vehicle communication relies on a radio transmitter located in each car that can transmit important information about the car's location and speed to other vehicles in the area. Each car equipped with a transmitter and a receiver can triangulate other vehicles in real time allowing the car or the driver to better react to what's going on around them. The advantage of using V2V or V2I is that we increase road safety, for example some applications of V2V are the Forward Collision Avoidance, the Blind Spot Warning, Lane Change Assist, Cooperative Adaptive Cruise Control, regarding the V2I applications in terms of road safety they are Red Light Running, Pedestrian Signal Assist, Rail Crossing.
Maneuver coordination messages allow vehicles to know what other vehicles intend to do ahead of time. Imagine a typical urban environment with several road users, vehicles, road infrastructures. All

the vehicles in this scenario may or may not have the capability to communicate wirelessly, having the V2X modules installed vehicles will communicate with each other however the wireless link directly or via cellular network to inform each other about their location, speed, heading direction and any relevant warnings for collision mitigation. Imagine a typical scenario, we have three cars approaching an intersection, one approaching from the left side, the other approaching from the right side and also an emergency vehicle that is approaching the intersection. We see everything from the viewpoint of the emergency vehicle, but this type of scenario is applicable to any intersection and any type of vehicle. Between the emergency vehicle and the vehicle that is approaching from the left there is a heavy obstruction. The left vehicle is called "obstructed vehicle" and the right vehicle that is called "visibility vehicle". Emergency vehicle has clear vision of the "visibility vehicle" while the left vehicle is misbehaving. This is a potential accident situation. Let's now enable V2X and let's see how this accident can be avoided. There is a target range that is calculated for these vehicles based on their speeds, for all vehicle we have 40 km/h so as soon as the target vehicle enters the target range which is shown with these white circles the emergency vehicle can detect the two other vehicles. The vehicle from the left side doesn't have a clear line of vision of the emergency vehicle so we have a number of transmissions to make sure that we can detect it reliably. The emergency vehicle starts tracking both the vehicles, the obstructed vehicle speed is not changing so as a result of this we see a warning message that is displayed on the dashboard and it starts braking and the accident is avoided.

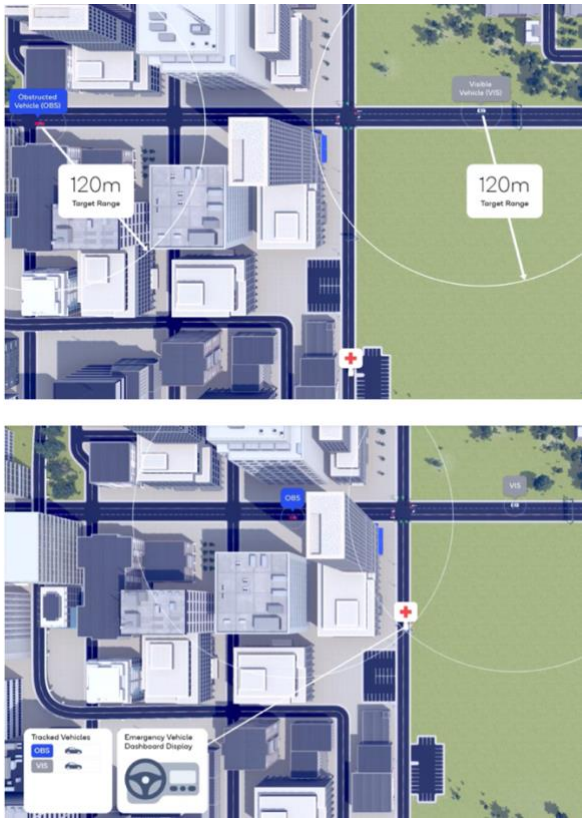

Figure 83: Application of vehicle communication

Let's think about a future where vehicles are able to communicate with each other. This vehicle-to-vehicle or V2V technology would provide crucial information to drivers when they need it and provide them with a warning of a potential crash. V2V provides a 360 degrees awareness of similarly equipped vehicles within a range of approximately 300 meters. This secure system keeps personal

information anonymous and does not track your vehicle. Drivers will receive warning that inform them of potential hazards through a visual display, seat vibration or tone; these are only warnings, the driver remains in control of the vehicle at all times. These warnings can help drivers to respond quickly to avoid potential crashes. For example, the intersection movement assist application warns drivers when it's not safe to enter an intersection, the "Do not Pass" application warns drivers when it's not safe to pass a slower moving vehicle, the "Emergency Electronic Brake Light" application notifies the driver when an out-of-sight vehicle, several cars ahead, is braking, the "Blind Spot Warning" application allows drivers to virtually see what's happening in his or her blind spots. These are just a few of the applications that may be available to help prevent car crashes in the near future.

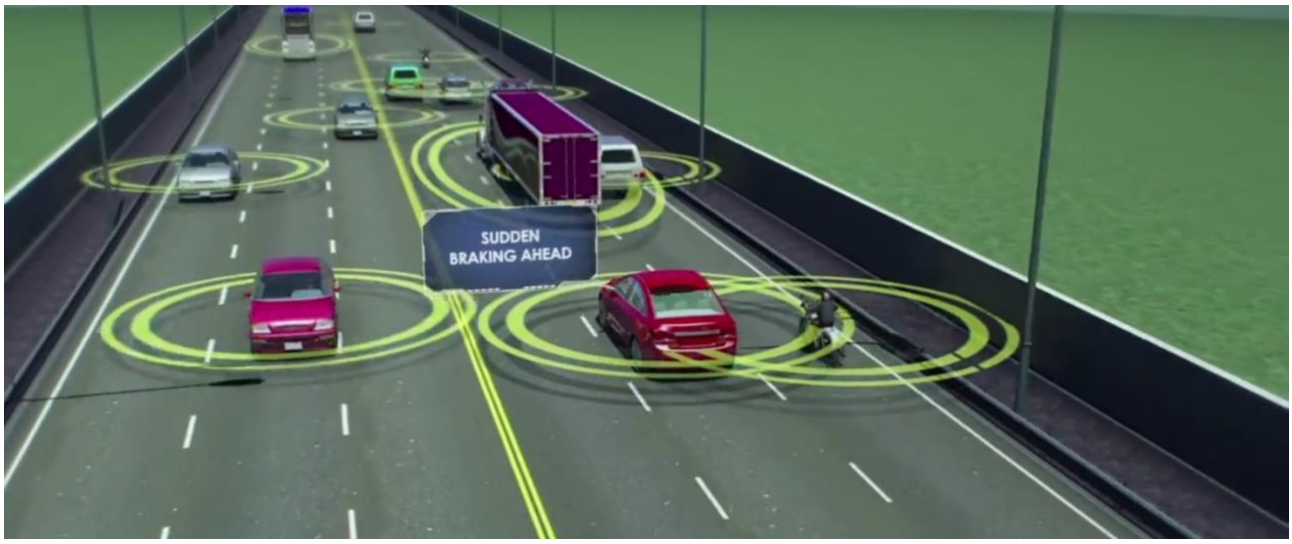

Figure 84: Emergency Electronic Brake Light Application

When vehicles are communicating to other vehicles, to the infrastructure, to the pedestrians or to mobile network we call vehicle to everything or V2X communication. V2X is a complete road safety and traffic efficiency solution that allows vehicles to communicate with the surrounding objects. In the connectivity landscape for the short-range communication vehicles and other road users often use direct communication between each other in an ad hoc manner whereas for the long-range communication or Internet of Things application they rely on 4G and 5G cellular networks. V2V, V2I and V2P operates in ITS bands (e.g. ITS 5.9GHz) independent of cellular network, while V2N operates in traditional mobile broadband licensed spectrum.

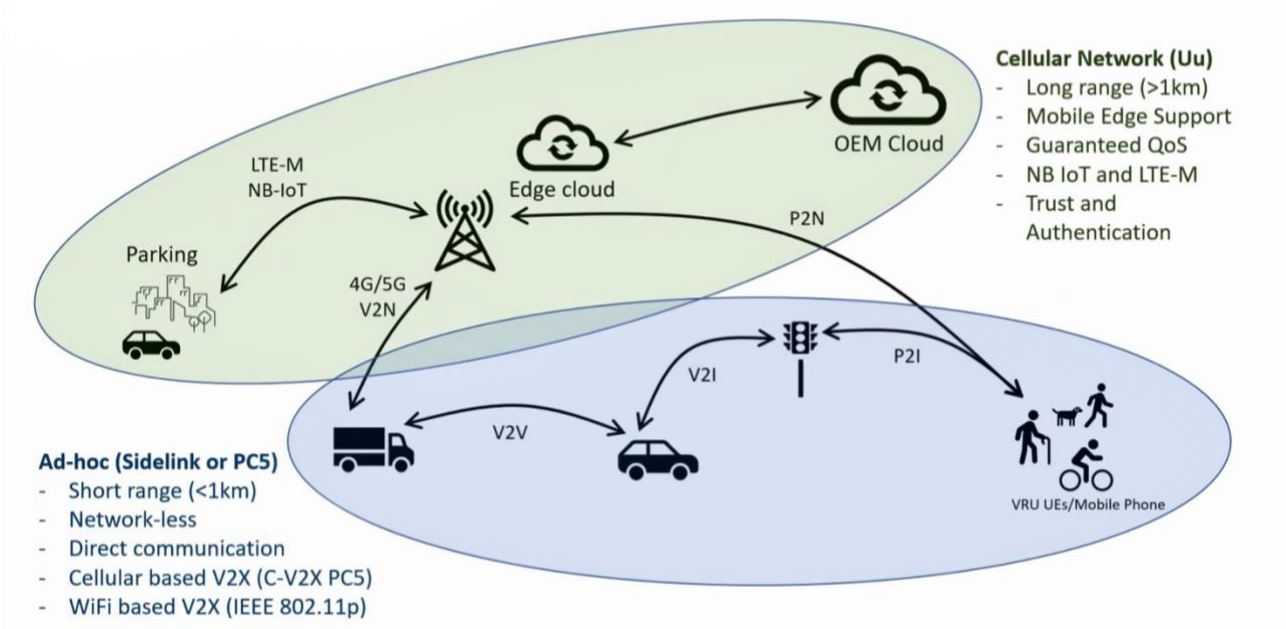

Figure 85: Different communication protocols

V2V uses a wireless protocol similar to Wi-fi called Dedicated Short-Range Communication or DSRC, for short. When DSRC is combined with GPS technology the result is low-cost V2V communication system that provides 360 degrees view of similarly equipped vehicles within communication range. Transmitted messages common to all vehicles include current GPS position, vehicle speed, acceleration and heading, vehicle control information such as transmission state, brake status and steering wheel angle as well as the vehicle's path history and path prediction.
FCC has authorized 75 MHz of spectrum (5.850-5.925) for DSRC. DSRC is a medium range communications service that operates in both Vehicle-to-Vehicle and Vehicle-to-infrastructure communication environments. "In V2V each vehicle broadcasts its core state information in a "Basic Safety Message" (BSM) nominally 10 times/sec. BSM is sent in 360 degrees pattern using IEEE 802.11p technology. Upon receipt of BSM, host vehicle's processor tracks each neighbor's trajectory, assesses threat to host vehicle, warns driver if threat becomes acute. Regarding the V2I roadside equipment (RSE), it can send Traveler Information message (TIM) on curve speed, height restriction, icy roads".[36] If you want to communicate to the next car through a cellular tower you are depending upon third part infrastructure that can delay as well as reliability issue, so the importance of having dedicated part like in DSRC is crucial. DSRC is a standard that is similar to Wi-Fi so not totally new standard and this is very important in the implementation phase, it is also characterized by low latency (less than 50 milliseconds), high data transfer (3-27 Mbps).

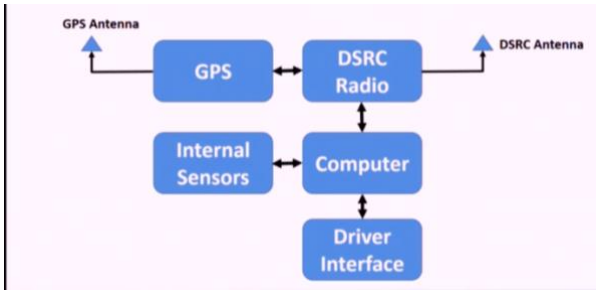

Figure 86: DSRC Protocol

"C-V2X is a unified platform which integrates short range, network-less, direct communications (LTE-V2X PC5 today) and long-range cellular network communications (LTE-V2X Uu today).

An additional safety benefit of C-V2X is the additional support for Vulnerable Road User (VRU) collision avoidance due to the integration of V2N (vehicle to network) and V2P (vehicle to pedestrian) functionality. This ensures that a VRU can become visible to vehicles in a first step via V2N utilizing smart phone applications carried by the VRU (and will not require the integration of smartphones with C-V2X functionality) and potentially later, in a second step, via direct V2P communication between vehicles and VRUs. C-V2X also has distinct capabilities that can be complementary to automated vehicle technologies and enable an enhanced experience and improved safety of highly automated driving. C-V2X offers a path in further 5G standardization to support significant increases in data transmission requirements that are paramount to automated vehicle technologies including the support for the exchange of:

- Sensor data for collective perception (e.g., video data)
- Control information for platoons from very close driving vehicles (only a few meters gap)
- Vehicle trajectories to prevent collisions (cooperative decision making).

These enhancements are not possible in other V2V and V2I technologies such as DSRC." [37] "Vehicular-to-everything (V2X) communication encompasses vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) messaging. V2X communication systems are expected to greatly improve road safety and traffic efficiency while better supporting autonomous driving. V2X promises to save lives directly by providing road hazard warnings to the driver and reducing collisions. The efficacy of V2X, however, is directly correlated to its adoption; the more vehicles enabled with V2X, the safer our roadways will be (and vice-versa). It is therefore of critical importance that V2X respects privacy in its design, not merely as a matter of legal compliance, but also to ensure consumer trust and mass adoption." [35] Another possible future application related with ITS is IoT. The easiest way to think of IoT is essentially imagine all the things that you use in your everyday life being connected. Automobiles are becoming the most sophisticated mobile device in the Internet of Things world, smart cars are now filled with rich services such as enhanced navigation, advanced safety features and integration between dashboards smartphones and wearables. All these systems and devices use a variety of wireless signals such as Bluetooth, LTE and Wi-Fi. Nowadays cars are like moving data centers, they have multitude of sensors, onboard computers that capture all sorts of information about the car being able to access that in real time. For example, IBM Watson IoT for automotive wants to optimize the connected car scenario, it is a unique software bundle that allows you to achieve unprecedent level of scale up to 13 million messages per second and even more importantly it is able to have concurrently millions of connected devices.

Conclusion

The aim of this work is to show the incredible opportunities offered by the intersection between ITS technologies and green mobility, in particular for public transport mobility, exploiting the case study of a fuel cell hybrid electric vehicle. The way in which this intersection takes place in this thesis consists in the development of a microcontroller able to acquire all the useful data to understand all the relevant technical aspects of the hybrid minibus. The obtained result is a STM32 based microcontroller able to obtain several information, like altitude, pressure, location, clock, temperature. In order to test all these sensors several tests have been carried out, exploiting an oled display in debug phase to show the results of computation and saving the acquired variables inside files created into an SD card folder. Regarding test on can bus, they have been carried out on board through an Arduino platform for time reasons. The innovative approach here, consists in the type of adopted microcontroller, instead of a very well-know, common-standard, and library friendly micro like Arduino I have used a stm32 based micro and a software called CubeIDE to set all the subsystems' interfaces, to achieve better performance and efficiency. As I have already mentioned in the previous chapters, there are possible future implementations, like developing a stm32 based interface with the can bus and also it would be interesting to realize server-micro interconnection through a sim card.

Bibliography

[1] Gli ITS per i PUMS nelle città italiane, TTS Italia, 2019;
[2] Libro Bianco sulla Politica dei Trasporti, Commissione Europea, 2001;
[3] https://it.wikipedia.org/wiki/Automatic_vehicle_monitoring;
[4] Le proposte di TTS Italia per la Smart Mobility, TTS Italia, 2019;
[5] United Nations Economic Commission for Europe (UNECE), Intelligent Transport Systems (ITS) for sustainable mobility, la fiaccola srl, 2012;
[6]https://www.who.int/news/item/22-09-2021-new-who-global-air-quality-guidelines-aim-to-save-millions-of-lives-from-air-pollution;
[7] Le applicazioni ITS per l'efficientamento della logistica, TTS Italia, 2021;
[8] Canale M., Course of "Techologies for autonomous vehicles", Politecnico di Torino;
[9] www.autostrade.it/assistenza-al-traffico/tutor.html;
[10] https://ec.europa.eu/environment/noise/europe_en.htm;
[11] Comunicazione della Commissione al Consiglio, al Parlamento europeo, al Comitato economico e sociale europeo e al Comitato delle regioni, Verso una strategia tematica sull'ambiente urbano. Bruxelles, 2004.
[12]http://geoportale.comune.torino.it/web/sites/default/files/mediafiles/pums_all1_linee_indirizzo_3.pdf;
[13] https://www.torinocitylab.it/it/experiment-to/circuito-smartroad;
[14]https://www.ansa.it/piemonte/notizie/2021/05/28/harvard-sceglie-torino-come-modello-di-smart-city_0086d61f-eebd-4211-9d2e-09b923530e2d.html;
[15] https://www.economyup.it/mobilita/smart-road/;
[16] EY mobility think tank 2020, La mobilità del possibile Volume 3, 2020;
[17]Strategia Nazionale Idrogeno, Linee Guida Preliminari, Ministero dello Sviluppo Economico, 2020;
[18 Hydrogen Roadmap Europe, A sustainable pathway for the European energy transition, Versione 1, Bietlot, 2019;
[19] Piano Nazionale di Sviluppo, Mobilità Idrogeno Italia, H2IT, 2019;
[20] IEA 2019, The Future of hydrogen, 2018;
[21] Fuelling Italy's Future, Come la transizione verso la mobilità a basso contenuto di carbonio rafforza l'economia, 2018;
[22] Il Mercato dei Sistemi Intelligenti di Trasporto in Italia: quadro attuale e prospettive, TTS Italia, 2016;
[23]https://temi.camera.it/leg17/post/il_recepimento_della_direttiva_dafi_sui_combustibili_alternativi.html?tema=temi/fonti_rinnovabili;
[24] Napoli G., Micari S., Dispenza G., Di Novo S., Antonucci V., Andaloro L., "Development of a fuel cell hybrid electric powertrain: A real case study on a Minibus application, International Journal of Hydrogen Energy 42, 2017, 28034-28047, Elsevier;

[25]Andaloro L., Arista A., Agnello G., Napoli G., Sergi F., Antonucci V., “Study and design of a hybrid electric vehicle”, International Journal of Hydrogen Energy, Volume 42, 2017, 3166-3184, Elsevier;
[26] Ferraro F., Andaloro L, Sergi F., Aloiso D., Dispenza G, Napoli G., Micari S., Brunaccini G., Randazzo N., Di Novo S., Antonucci V., “Electrochemical energy storage mitigating impact of electric vehicle on the electric grid: two Italian case studies”, 2017.
[27]Napoli G., Sergi F., Randazzo N., Micari S., Dispenza G., Antonucci V., Di Novo S., Andaloro L., “Development of a solar powered hydrogen fueling station in smart cities applications”, International journal of hydrogen energy,42, Elsevier, 2017;
[28]Violante M., “Operating systems for embedded systems”, Politecnico di Torino.
[29] https://www.cypress.com/file/132486/download;
[30] Slides on USART, Microchip, 2001;
[31] Panzieri S., Sciavicco L., Course of “Controllo digitale”, Università degli studi Roma Tre;
[32] https://datasheets.maximintegrated.com/en/ds/DS3231.pdf
[33] https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf
[34]https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf
[35]http://datalogger.pbworks.com/w/file/fetch/89507207/Datalogger%20-%20SD%20Memory%20Reader%20Datasheet.pdf
[36] Roadway Safety Institute, Human-centered solutions to advance roadway safety, Department of Electrical Engineering, University of Minnesota Duluth;
[37] Connectivity Standards in the Automotive Industry, J. Springer, 5GAA Automotive Association;
[38] Privacy by Design Aspects of C-V2X, 5GAA Automotive Association.
[39] https://controllerstech.com

Appendix

Rtc-Oled main

```
/* USER CODE END Header */
/* Includes ------------------------------------------------------------------*/
#include "main.h"
/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
#include "stm32f4xx_hal.h"
//#include "fonts.h"
#include "ssd1306.h"
#include <stdbool.h>
#include "stdio.h"
/* USER CODE END Includes */
/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN PTD */
/* USER CODE END PTD */
/* Private define ------------------------------------------------------------*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */
/* Private macro -------------------------------------------------------------*/
/* USER CODE BEGIN PM */
/* USER CODE END PM */
/* Private variables ---------------------------------------------------------*/
I2C_HandleTypeDef hi2c1;
/* USER CODE BEGIN PV */
/* USER CODE END PV */
/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_I2C1_Init(void);
/* USER CODE BEGIN PFP */
/* USER CODE END PFP */
/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
#define DS3231_ADDRESS 0xD0
// Convert normal decimal numbers to binary coded decimal
uint8_t decToBcd(int val)
```

```
{
  return (uint8_t)( (val/10*16) + (val%10) );
}
// Convert binary coded decimal to normal decimal numbers
int bcdToDec(uint8_t val)
{
  return (int)( (val/16*10) + (val%16) );
}
typedef struct {
        uint8_t seconds;
        uint8_t minutes;
        uint8_t hour;
        uint8_t dayofweek;
        uint8_t dayofmonth;
        uint8_t month;
        uint8_t year;
} TIME;
TIME time;
// function to set time
uint8_t set_time[7];
//Function to set the time data and write them in the DS3231 address through i2c
void Set_Time (uint8_t sec, uint8_t min, uint8_t hour, uint8_t dow, uint8_t dom, uint8_t month, uint8_t year)
{
        set_time[0] = decToBcd(sec);
        set_time[1] = decToBcd(min);
        set_time[2] = decToBcd(hour);
        set_time[3] = decToBcd(dow);
        set_time[4] = decToBcd(dom);
        set_time[5] = decToBcd(month);
        set_time[6] = decToBcd(year);

        HAL_I2C_Mem_Write(&hi2c1, DS3231_ADDRESS, 0x00, 1, set_time, 7, 100);
}
//function to get the time data previously collected and stored in the DS3231 address
void Get_Time (void)
{
        uint8_t get_time[7];
        HAL_I2C_Mem_Read(&hi2c1, DS3231_ADDRESS, 0x00, 1, get_time, 7, 1000);
        time.seconds = bcdToDec(get_time[0]);
        time.minutes = bcdToDec(get_time[1]);
```

```
	time.hour = bcdToDec(get_time[2]);
	time.dayofweek = bcdToDec(get_time[3]);
	time.dayofmonth = bcdToDec(get_time[4]);
	time.month = bcdToDec(get_time[5]);
	time.year = bcdToDec(get_time[6]);
}
// Array for RTC clock data
char buffer[15];
/* USER CODE END 0 */
/**
  * @brief  The application entry point.
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */
  /* USER CODE END 1 */
  /* MCU Configuration--------------------------------------------------------*/
  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();
  /* USER CODE BEGIN Init */
  /* USER CODE END Init */
  /* Configure the system clock */
  SystemClock_Config();
  /* USER CODE BEGIN SysInit */
  /* USER CODE END SysInit */
  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_I2C1_Init();
  /* USER CODE BEGIN 2 */
  Set_Time(0, 48, 8, 5, 10, 12, 21);
  /* USER CODE END 2 */
  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */
    /* USER CODE BEGIN 3 */
		Get_Time();
		ssd1306_Fill(Black);
		sprintf (buffer, "%02d-%02d-20%02d", time.dayofmonth, time.month, time.year);
```

```
            ssd1306_SetCursor(0, 0);
            ssd1306_WriteString(buffer, Font_6x8, White);
            sprintf (buffer, "%02d:%02d:%02d", time.hour, time.minutes, time.seconds);
            ssd1306_SetCursor(72, 0);
            ssd1306_WriteString(buffer, Font_6x8, White);
            ssd1306_UpdateScreen();
            HAL_Delay(500);
  }
  /* USER CODE END 3 */
}
/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

  /** Configure the main internal regulator output voltage
  */
  __HAL_RCC_PWR_CLK_ENABLE();
  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
  /** Initializes the RCC Oscillators according to the specified parameters
  * in the RCC_OscInitTypeDef structure.
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
  RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
  RCC_OscInitStruct.PLL.PLLM = 4;
  RCC_OscInitStruct.PLL.PLLN = 180;
  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
  RCC_OscInitStruct.PLL.PLLQ = 4;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    Error_Handler();
  }
  /** Activate the Over-Drive mode
  */
  if (HAL_PWREx_EnableOverDrive() != HAL_OK)
  {
```

```
    Error_Handler();
  }
  /** Initializes the CPU, AHB and APB buses clocks
  */
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                              |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != HAL_OK)
  {
    Error_Handler();
  }
}
/**
  * @brief I2C1 Initialization Function
  * @param None
  * @retval None
  */
static void MX_I2C1_Init(void)
{
  /* USER CODE BEGIN I2C1_Init 0 */
  /* USER CODE END I2C1_Init 0 */
  /* USER CODE BEGIN I2C1_Init 1 */
  /* USER CODE END I2C1_Init 1 */
  hi2c1.Instance = I2C1;
  hi2c1.Init.ClockSpeed = 400000;
  hi2c1.Init.DutyCycle = I2C_DUTYCYCLE_2;
  hi2c1.Init.OwnAddress1 = 0;
  hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
  hi2c1.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
  hi2c1.Init.OwnAddress2 = 0;
  hi2c1.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
  hi2c1.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
  if (HAL_I2C_Init(&hi2c1) != HAL_OK)
  {
    Error_Handler();
  }
  /** Configure Analogue filter
  */
```

```
  if (HAL_I2CEx_ConfigAnalogFilter(&hi2c1, I2C_ANALOGFILTER_ENABLE) != HAL_OK)
  {
    Error_Handler();
  }
  /** Configure Digital filter
  */
  if (HAL_I2CEx_ConfigDigitalFilter(&hi2c1, 0) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN I2C1_Init 2 */
  /* USER CODE END I2C1_Init 2 */
}
/**
  * @brief GPIO Initialization Function
  * @param None
  * @retval None
  */
static void MX_GPIO_Init(void)
{
  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOH_CLK_ENABLE();
  __HAL_RCC_GPIOA_CLK_ENABLE();
  __HAL_RCC_GPIOB_CLK_ENABLE();


}
/* USER CODE BEGIN 4 */
/* USER CODE END 4 */
/**
  * @brief  This function is executed in case of error occurrence.
  * @retval None
  */
void Error_Handler(void)
{
  /* USER CODE BEGIN Error_Handler_Debug */
  /* User can add his own implementation to report the HAL error return state */
  __disable_irq();
  while (1)
  {
  }
  /* USER CODE END Error_Handler_Debug */
}
#ifdef  USE_FULL_ASSERT
```

```
/**
  * @brief  Reports the name of the source file and the source line number
  *         where the assert_param error has occurred.
  * @param  file: pointer to the source file name
  * @param  line: assert_param error line source number
  * @retval None
  */
void assert_failed(uint8_t *file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
  /* User can add his own implementation to report the file name and line number,
     ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
  /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */
```

Rtc-Oled libraries

SSD1306.c [39]

```
#include "ssd1306.h"
#include <math.h>
#include <stdlib.h>
#include <string.h>  // For memcpy
#if defined(SSD1306_USE_I2C)
void ssd1306_Reset(void) {
    /* for I2C - do nothing */
}
// Send a byte to the command register
void ssd1306_WriteCommand(uint8_t byte) {
    HAL_I2C_Mem_Write(&SSD1306_I2C_PORT, SSD1306_I2C_ADDR, 0x00, 1, &byte, 1, HAL_MAX_DELAY);
}
// Send data
void ssd1306_WriteData(uint8_t* buffer, size_t buff_size) {
    HAL_I2C_Mem_Write(&SSD1306_I2C_PORT, SSD1306_I2C_ADDR, 0x40, 1, buffer, buff_size, HAL_MAX_DELAY);
}
#elif defined(SSD1306_USE_SPI)
void ssd1306_Reset(void) {
    // CS = High (not selected)
    HAL_GPIO_WritePin(SSD1306_CS_Port, SSD1306_CS_Pin, GPIO_PIN_SET);

    // Reset the OLED
```

```
    HAL_GPIO_WritePin(SSD1306_Reset_Port, SSD1306_Reset_Pin, GPIO_PIN_RESET);
    HAL_Delay(10);
    HAL_GPIO_WritePin(SSD1306_Reset_Port, SSD1306_Reset_Pin, GPIO_PIN_SET);
    HAL_Delay(10);
}
// Send a byte to the command register
void ssd1306_WriteCommand(uint8_t byte) {
    HAL_GPIO_WritePin(SSD1306_CS_Port, SSD1306_CS_Pin, GPIO_PIN_RESET); // select OLED
    HAL_GPIO_WritePin(SSD1306_DC_Port, SSD1306_DC_Pin, GPIO_PIN_RESET); // command
    HAL_SPI_Transmit(&SSD1306_SPI_PORT, (uint8_t *) &byte, 1, HAL_MAX_DELAY);
    HAL_GPIO_WritePin(SSD1306_CS_Port, SSD1306_CS_Pin, GPIO_PIN_SET); // un-select OLED
}

// Send data
void ssd1306_WriteData(uint8_t* buffer, size_t buff_size) {
    HAL_GPIO_WritePin(SSD1306_CS_Port, SSD1306_CS_Pin, GPIO_PIN_RESET); // select OLED
    HAL_GPIO_WritePin(SSD1306_DC_Port, SSD1306_DC_Pin, GPIO_PIN_SET); // data
    HAL_SPI_Transmit(&SSD1306_SPI_PORT, buffer, buff_size, HAL_MAX_DELAY);
    HAL_GPIO_WritePin(SSD1306_CS_Port, SSD1306_CS_Pin, GPIO_PIN_SET); // un-select OLED
}

#else
#error "You should define SSD1306_USE_SPI or SSD1306_USE_I2C macro"
#endif
// Screenbuffer
static uint8_t SSD1306_Buffer[SSD1306_BUFFER_SIZE];
// Screen object
static SSD1306_t SSD1306;
/* Fills the Screenbuffer with values from a given buffer of a fixed length */
SSD1306_Error_t ssd1306_FillBuffer(uint8_t* buf, uint32_t len) {
    SSD1306_Error_t ret = SSD1306_ERR;
    if (len <= SSD1306_BUFFER_SIZE) {
        memcpy(SSD1306_Buffer,buf,len);
        ret = SSD1306_OK;
    }
    return ret;
}
// Initialize the oled screen
void ssd1306_Init(void) {
    // Reset OLED
```

```
    ssd1306_Reset();
    // Wait for the screen to boot
    HAL_Delay(100);
    // Init OLED
    ssd1306_SetDisplayOn(0); //display off
    ssd1306_WriteCommand(0x20); //Set Memory Addressing Mode
    ssd1306_WriteCommand(0x00); // 00b,Horizontal Addressing Mode; 01b,Vertical Addressing Mode;
                                // 10b,Page Addressing Mode (RESET); 11b,Invalid
    ssd1306_WriteCommand(0xB0); //Set Page Start Address for Page Addressing Mode,0-7
#ifdef SSD1306_MIRROR_VERT
    ssd1306_WriteCommand(0xC0); // Mirror vertically
#else
    ssd1306_WriteCommand(0xC8); //Set COM Output Scan Direction
#endif
    ssd1306_WriteCommand(0x00); //---set low column address
    ssd1306_WriteCommand(0x10); //---set high column address
    ssd1306_WriteCommand(0x40); //--set start line address - CHECK
    ssd1306_SetContrast(0xFF);
#ifdef SSD1306_MIRROR_HORIZ
    ssd1306_WriteCommand(0xA0); // Mirror horizontally
#else
    ssd1306_WriteCommand(0xA1); //--set segment re-map 0 to 127 - CHECK
#endif
#ifdef SSD1306_INVERSE_COLOR
    ssd1306_WriteCommand(0xA7); //--set inverse color
#else
    ssd1306_WriteCommand(0xA6); //--set normal color
#endif
// Set multiplex ratio.
#if (SSD1306_HEIGHT == 128)
    // Found in the Luma Python lib for SH1106.
    ssd1306_WriteCommand(0xFF);
#else
    ssd1306_WriteCommand(0xA8); //--set multiplex ratio(1 to 64) - CHECK
#endif
#if (SSD1306_HEIGHT == 32)
    ssd1306_WriteCommand(0x1F); //
#elif (SSD1306_HEIGHT == 64)
    ssd1306_WriteCommand(0x3F); //
#elif (SSD1306_HEIGHT == 128)
```

```
    ssd1306_WriteCommand(0x3F); // Seems to work for 128px high displays too.
#else
#error "Only 32, 64, or 128 lines of height are supported!"
#endif

    ssd1306_WriteCommand(0xA4); //0xa4,Output follows RAM content;0xa5,Output ignores RAM content
    ssd1306_WriteCommand(0xD3); //-set display offset - CHECK
    ssd1306_WriteCommand(0x00); //-not offset
    ssd1306_WriteCommand(0xD5); //--set display clock divide ratio/oscillator frequency
    ssd1306_WriteCommand(0xF0); //--set divide ratio
    ssd1306_WriteCommand(0xD9); //--set pre-charge period
    ssd1306_WriteCommand(0x22); //
    ssd1306_WriteCommand(0xDA); //--set com pins hardware configuration - CHECK
#if (SSD1306_HEIGHT == 32)
    ssd1306_WriteCommand(0x02);
#elif (SSD1306_HEIGHT == 64)
    ssd1306_WriteCommand(0x12);
#elif (SSD1306_HEIGHT == 128)
    ssd1306_WriteCommand(0x12);
#else
#error "Only 32, 64, or 128 lines of height are supported!"
#endif
    ssd1306_WriteCommand(0xDB); //--set vcomh
    ssd1306_WriteCommand(0x20); //0x20,0.77xVcc
    ssd1306_WriteCommand(0x8D); //--set DC-DC enable
    ssd1306_WriteCommand(0x14); //
    ssd1306_SetDisplayOn(1); //--turn on SSD1306 panel
    // Clear screen
    ssd1306_Fill(Black);
    // Flush buffer to screen
    ssd1306_UpdateScreen();
    // Set default values for screen object
    SSD1306.CurrentX = 0;
    SSD1306.CurrentY = 0;
    SSD1306.Initialized = 1;
}
// Fill the whole screen with the given color
void ssd1306_Fill(SSD1306_COLOR color) {
    /* Set memory */
    uint32_t i;
```

```
    for(i = 0; i < sizeof(SSD1306_Buffer); i++) {
        SSD1306_Buffer[i] = (color == Black) ? 0x00 : 0xFF;
    }
}
// Write the screenbuffer with changed to the screen
void ssd1306_UpdateScreen(void) {
    // Write data to each page of RAM. Number of pages
    // depends on the screen height:
    //
    //  * 32px   ==  4 pages
    //  * 64px   ==  8 pages
    //  * 128px  ==  16 pages
    for(uint8_t i = 0; i < SSD1306_HEIGHT/8; i++) {
        ssd1306_WriteCommand(0xB0 + i); // Set the current RAM page address.
        ssd1306_WriteCommand(0x00);
        ssd1306_WriteCommand(0x10);
ssd1306_WriteData(&SSD1306_Buffer[SSD1306_WIDTH*i],SSD1306_WIDTH);
    }
}
//   Draw one pixel in the screenbuffer
//   X => X Coordinate
//   Y => Y Coordinate
//   color => Pixel color
void ssd1306_DrawPixel(uint8_t x, uint8_t y, SSD1306_COLOR color) {
    if(x >= SSD1306_WIDTH || y >= SSD1306_HEIGHT) {
        // Don't write outside the buffer
        return;
    }
    // Check if pixel should be inverted
    if(SSD1306.Inverted) {
        color = (SSD1306_COLOR)!color;
    }
    // Draw in the right color
    if(color == White) {
        SSD1306_Buffer[x + (y / 8) * SSD1306_WIDTH] |= 1 << (y % 8);
    } else {
        SSD1306_Buffer[x + (y / 8) * SSD1306_WIDTH] &= ~(1 << (y % 8));
    }
}
// Draw 1 char to the screen buffer
```

```
// ch       => char om weg te schrijven
// Font     => Font waarmee we gaan schrijven
// color    => Black or White
char ssd1306_WriteChar(char ch, FontDef Font, SSD1306_COLOR color) {
    uint32_t i, b, j;
    // Check if character is valid
    if (ch < 32 || ch > 126)
        return 0;
    // Check remaining space on current line
    if (SSD1306_WIDTH < (SSD1306.CurrentX + Font.FontWidth) ||
        SSD1306_HEIGHT < (SSD1306.CurrentY + Font.FontHeight))
    {
        // Not enough space on current line
        return 0;
    }
    // Use the font to write
    for(i = 0; i < Font.FontHeight; i++) {
        b = Font.data[(ch - 32) * Font.FontHeight + i];
        for(j = 0; j < Font.FontWidth; j++) {
            if((b << j) & 0x8000)  {
                ssd1306_DrawPixel(SSD1306.CurrentX + j, (SSD1306.CurrentY + i), (SSD1306_COLOR) color);
            } else {
                ssd1306_DrawPixel(SSD1306.CurrentX + j, (SSD1306.CurrentY + i), (SSD1306_COLOR)!color);
            }
        }
    }
    // The current space is now taken
    SSD1306.CurrentX += Font.FontWidth;
    // Return written char for validation
    return ch;
}
// Write full string to screenbuffer
char ssd1306_WriteString(char* str, FontDef Font, SSD1306_COLOR color) {
    // Write until null-byte
    while (*str) {
        if (ssd1306_WriteChar(*str, Font, color) != *str) {
            // Char could not be written
            return *str;
        }
        // Next char
```

```
        str++;
    }
    // Everything ok
    return *str;
}
// Position the cursor
void ssd1306_SetCursor(uint8_t x, uint8_t y) {
    SSD1306.CurrentX = x;
    SSD1306.CurrentY = y;
}
// Draw line by Bresenhem's algorithm
void ssd1306_Line(uint8_t x1, uint8_t y1, uint8_t x2, uint8_t y2, SSD1306_COLOR color) {
  int32_t deltaX = abs(x2 - x1);
  int32_t deltaY = abs(y2 - y1);
  int32_t signX = ((x1 < x2) ? 1 : -1);
  int32_t signY = ((y1 < y2) ? 1 : -1);
  int32_t error = deltaX - deltaY;
  int32_t error2;
  ssd1306_DrawPixel(x2, y2, color);
    while((x1 != x2) || (y1 != y2))
    {
    ssd1306_DrawPixel(x1, y1, color);
    error2 = error * 2;
    if(error2 > -deltaY)
    {
      error -= deltaY;
      x1 += signX;
    }
    else
    {
    /*nothing to do*/
    }

    if(error2 < deltaX)
    {
      error += deltaX;
      y1 += signY;
    }
    else
    {
```

```
    /*nothing to do*/
    }
  }
  return;
}
//Draw polyline
void ssd1306_Polyline(const SSD1306_VERTEX *par_vertex, uint16_t par_size, SSD1306_COLOR color) {
  uint16_t i;
  if(par_vertex != 0){
    for(i = 1; i < par_size; i++){
      ssd1306_Line(par_vertex[i - 1].x, par_vertex[i - 1].y, par_vertex[i].x, par_vertex[i].y, color);
    }
  }
  else
  {
    /*nothing to do*/
  }
  return;
}
/*Convert Degrees to Radians*/
static float ssd1306_DegToRad(float par_deg) {
    return par_deg * 3.14 / 180.0;
}
/*Normalize degree to [0;360]*/
static uint16_t ssd1306_NormalizeTo0_360(uint16_t par_deg) {
  uint16_t loc_angle;
  if(par_deg <= 360)
  {
    loc_angle = par_deg;
  }
  else
  {
    loc_angle = par_deg % 360;
    loc_angle = ((par_deg != 0)?par_deg:360);
  }
  return loc_angle;
}
/*DrawArc. Draw angle is beginning from 4 quart of trigonometric circle (3pi/2)
 * start_angle in degree
 * sweep in degree
```

```
 */
void ssd1306_DrawArc(uint8_t x, uint8_t y, uint8_t radius, uint16_t start_angle, uint16_t sweep, SSD1306_COLOR color)
{
    #define CIRCLE_APPROXIMATION_SEGMENTS 36
    float approx_degree;
    uint32_t approx_segments;
    uint8_t xp1,xp2;
    uint8_t yp1,yp2;
    uint32_t count = 0;
    uint32_t loc_sweep = 0;
    float rad;
    loc_sweep = ssd1306_NormalizeTo0_360(sweep);

    count = (ssd1306_NormalizeTo0_360(start_angle) * CIRCLE_APPROXIMATION_SEGMENTS) / 360;
    approx_segments = (loc_sweep * CIRCLE_APPROXIMATION_SEGMENTS) / 360;
    approx_degree = loc_sweep / (float)approx_segments;
    while(count < approx_segments)
    {
        rad = ssd1306_DegToRad(count*approx_degree);
        xp1 = x + (int8_t)(sin(rad)*radius);
        yp1 = y + (int8_t)(cos(rad)*radius);
        count++;
        if(count != approx_segments)
        {
            rad = ssd1306_DegToRad(count*approx_degree);
        }
        else
        {
            rad = ssd1306_DegToRad(loc_sweep);
        }
        xp2 = x + (int8_t)(sin(rad)*radius);
        yp2 = y + (int8_t)(cos(rad)*radius);
        ssd1306_Line(xp1,yp1,xp2,yp2,color);
    }

    return;
}
//Draw circle by Bresenhem's algorithm
void ssd1306_DrawCircle(uint8_t par_x,uint8_t par_y,uint8_t par_r,SSD1306_COLOR par_color) {
  int32_t x = -par_r;
```

```
  int32_t y = 0;
  int32_t err = 2 - 2 * par_r;
  int32_t e2;
  if (par_x >= SSD1306_WIDTH || par_y >= SSD1306_HEIGHT) {
    return;
  }
    do {
      ssd1306_DrawPixel(par_x - x, par_y + y, par_color);
      ssd1306_DrawPixel(par_x + x, par_y + y, par_color);
      ssd1306_DrawPixel(par_x + x, par_y - y, par_color);
      ssd1306_DrawPixel(par_x - x, par_y - y, par_color);
        e2 = err;
        if (e2 <= y) {
            y++;
            err = err + (y * 2 + 1);
            if(-x == y && e2 <= x) {
              e2 = 0;
            }
            else
            {
              /*nothing to do*/
            }
        }
        else
        {
          /*nothing to do*/
        }
        if(e2 > x) {
          x++;
          err = err + (x * 2 + 1);
        }
        else
        {
          /*nothing to do*/
        }
    } while(x <= 0);
    return;
}
//Draw rectangle
void ssd1306_DrawRectangle(uint8_t x1, uint8_t y1, uint8_t x2, uint8_t y2, SSD1306_COLOR color) {
```

```
  ssd1306_Line(x1,y1,x2,y1,color);
  ssd1306_Line(x2,y1,x2,y2,color);
  ssd1306_Line(x2,y2,x1,y2,color);
  ssd1306_Line(x1,y2,x1,y1,color);
  return;
}
void ssd1306_SetContrast(const uint8_t value) {
    const uint8_t kSetContrastControlRegister = 0x81;
    ssd1306_WriteCommand(kSetContrastControlRegister);
    ssd1306_WriteCommand(value);
}
void ssd1306_SetDisplayOn(const uint8_t on) {
    uint8_t value;
    if (on) {
        value = 0xAF;   // Display on
        SSD1306.DisplayOn = 1;
    } else {
        value = 0xAE;   // Display off
        SSD1306.DisplayOn = 0;
    }
    ssd1306_WriteCommand(value);
}
uint8_t ssd1306_GetDisplayOn() {
    return SSD1306.DisplayOn;
}
```

SSD1306.h [39]

```
* https://github.com/afiskon/stm32-ssd1306
 */
#ifndef __SSD1306_H__
#define __SSD1306_H__
#include <stddef.h>
#include <_ansi.h>
_BEGIN_STD_C
#include "ssd1306_conf.h"
#if defined(STM32F0)
#include "stm32f0xx_hal.h"
#elif defined(STM32F1)
#include "stm32f1xx_hal.h"
#elif defined(STM32F4)
#include "stm32f4xx_hal.h"
#include "stm32f4xx_hal_gpio.h"
#elif defined(STM32L0)
#include "stm32l0xx_hal.h"
#elif defined(STM32L1)
#include "stm32l1xx_hal.h"
#elif defined(STM32L4)
#include "stm32l4xx_hal.h"
#elif defined(STM32F3)
```

```
#include "stm32f3xx_hal.h"
#elif defined(STM32H7)
#include "stm32h7xx_hal.h"
#elif defined(STM32F7)
#include "stm32f7xx_hal.h"
#elif defined(STM32G0)
#include "stm32g0xx_hal.h"
#else
#error "SSD1306 library was tested only on STM32F0, STM32F1, STM32F3, STM32F4, STM32F7, STM32L0,
STM32L1, STM32L4, STM32H7, STM32G0 MCU families. Please modify ssd1306.h if you know what you are doing.
Also please send a pull request if it turns out the library works on other MCU's as well!"
#endif
#include "ssd1306_fonts.h"
/* vvv I2C config vvv */
#ifndef SSD1306_I2C_PORT
#define SSD1306_I2C_PORT        hi2c1
#endif
#ifndef SSD1306_I2C_ADDR
#define SSD1306_I2C_ADDR        (0x3C << 1)
#endif
/* ^^^ I2C config ^^^ */
/* vvv SPI config vvv */
#ifndef SSD1306_SPI_PORT
#define SSD1306_SPI_PORT        hspi2
#endif
#ifndef SSD1306_CS_Port
#define SSD1306_CS_Port         GPIOB
#endif
#ifndef SSD1306_CS_Pin
#define SSD1306_CS_Pin          GPIO_PIN_12
#endif
#ifndef SSD1306_DC_Port
#define SSD1306_DC_Port         GPIOB
#endif
#ifndef SSD1306_DC_Pin
#define SSD1306_DC_Pin          GPIO_PIN_14
#endif
#ifndef SSD1306_Reset_Port
#define SSD1306_Reset_Port      GPIOA
#endif
```

```
#ifndef SSD1306_Reset_Pin
#define SSD1306_Reset_Pin       GPIO_PIN_8
#endif
/* ^^^ SPI config ^^^ */
#if defined(SSD1306_USE_I2C)
extern I2C_HandleTypeDef SSD1306_I2C_PORT;
#elif defined(SSD1306_USE_SPI)
extern SPI_HandleTypeDef SSD1306_SPI_PORT;
#else
#error "You should define SSD1306_USE_SPI or SSD1306_USE_I2C macro!"
#endif
// SSD1306 OLED height in pixels
#ifndef SSD1306_HEIGHT
#define SSD1306_HEIGHT          64
#endif
// SSD1306 width in pixels
#ifndef SSD1306_WIDTH
#define SSD1306_WIDTH           128
#endif
#ifndef SSD1306_BUFFER_SIZE
#define SSD1306_BUFFER_SIZE   SSD1306_WIDTH * SSD1306_HEIGHT / 8
#endif
// Enumeration for screen colors
typedef enum {
    Black = 0x00, // Black color, no pixel
    White = 0x01  // Pixel is set. Color depends on OLED
} SSD1306_COLOR;

typedef enum {
    SSD1306_OK = 0x00,
    SSD1306_ERR = 0x01  // Generic error.
} SSD1306_Error_t;

// Struct to store transformations
typedef struct {
    uint16_t CurrentX;
    uint16_t CurrentY;
    uint8_t Inverted;
    uint8_t Initialized;
    uint8_t DisplayOn;
```

```
} SSD1306_t;
typedef struct {
    uint8_t x;
    uint8_t y;
} SSD1306_VERTEX;
// Procedure definitions
void ssd1306_Init(void);
void ssd1306_Fill(SSD1306_COLOR color);
void ssd1306_UpdateScreen(void);
void ssd1306_DrawPixel(uint8_t x, uint8_t y, SSD1306_COLOR color);
char ssd1306_WriteChar(char ch, FontDef Font, SSD1306_COLOR color);
char ssd1306_WriteString(char* str, FontDef Font, SSD1306_COLOR color);
void ssd1306_SetCursor(uint8_t x, uint8_t y);
void ssd1306_Line(uint8_t x1, uint8_t y1, uint8_t x2, uint8_t y2, SSD1306_COLOR color);
void ssd1306_DrawArc(uint8_t x, uint8_t y, uint8_t radius, uint16_t start_angle, uint16_t sweep, SSD1306_COLOR color);
void ssd1306_DrawCircle(uint8_t par_x, uint8_t par_y, uint8_t par_r, SSD1306_COLOR color);
void ssd1306_Polyline(const SSD1306_VERTEX *par_vertex, uint16_t par_size, SSD1306_COLOR color);
void ssd1306_DrawRectangle(uint8_t x1, uint8_t y1, uint8_t x2, uint8_t y2, SSD1306_COLOR color);
/**
 * @brief Sets the contrast of the display.
 * @param[in] value contrast to set.
 * @note Contrast increases as the value increases.
 * @note RESET = 7Fh.
 */
void ssd1306_SetContrast(const uint8_t value);
/**
 * @brief Set Display ON/OFF.
 * @param[in] on 0 for OFF, any for ON.
 */
void ssd1306_SetDisplayOn(const uint8_t on);
/**
 * @brief Reads DisplayOn state.
 * @return  0: OFF.
 *          1: ON.
 */
uint8_t ssd1306_GetDisplayOn();
// Low-level procedures
void ssd1306_Reset(void);
void ssd1306_WriteCommand(uint8_t byte);
void ssd1306_WriteData(uint8_t* buffer, size_t buff_size);
```

```
SSD1306_Error_t ssd1306_FillBuffer(uint8_t* buf, uint32_t len);

_END_STD_C

#endif // __SSD1306_H__
```

BMP180 main

```
/* USER CODE END Header */
/* Includes ------------------------------------------------------------------*/
#include "main.h"
/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
#include "BMP180.h"
#include "string.h"
#include "stdio.h"
#include "ssd1306.h"
/* USER CODE END Includes */
/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN PTD */
/* USER CODE END PTD */
/* Private define ------------------------------------------------------------*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */
/* Private macro -------------------------------------------------------------*/
/* USER CODE BEGIN PM */
/* USER CODE END PM */
```

```
/* Private variables ---------------------------------------------------------*/
I2C_HandleTypeDef hi2c1;
/* USER CODE BEGIN PV */
/* USER CODE END PV */
/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_I2C1_Init(void);
/* USER CODE BEGIN PFP */
/* USER CODE END PFP */
/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
float Temperature;
float Pressure;
float Altitude;
char Temperature1[10];
char Pressure1[10];
char Altitude1[10];
/* USER CODE END 0 */
/**
  * @brief  The application entry point.
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */
  /* USER CODE END 1 */
  /* MCU Configuration--------------------------------------------------------*/
  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();
  /* USER CODE BEGIN Init */
  /* USER CODE END Init */
  /* Configure the system clock */
  SystemClock_Config();
  /* USER CODE BEGIN SysInit */
  /* USER CODE END SysInit */
  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_I2C1_Init();
  /* USER CODE BEGIN 2 */
  BMP180_Start();
```

```
ssd1306_Init();
ssd1306_SetCursor(35,0);
ssd1306_WriteString ("BMP180", Font_11x18, White);
ssd1306_SetCursor (10,20);
ssd1306_WriteString  ("Barometric", Font_11x18, White);
ssd1306_SetCursor(30,40);
ssd1306_WriteString  ("Sensor", Font_11x18, White);
ssd1306_SetCursor (20,40);
ssd1306_UpdateScreen(); //display
HAL_Delay(2000);
/* USER CODE END 2 */
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
  /* USER CODE END WHILE */
  /* USER CODE BEGIN 3 */
        Temperature=BMP180_GetTemp();
        Pressure=BMP180_GetPress(0);
        Altitude=BMP180_GetAlt(0);
        ssd1306_Fill(Black);
        ssd1306_SetCursor(0,0);
        ssd1306_WriteString("BMP180", Font_7x10, White);
        ssd1306_SetCursor (0,0);
        ssd1306_WriteString("Temperature", Font_7x10, White);
        ssd1306_SetCursor (0,12);
        sprintf(Temperature1, "%.2f", Temperature);
        ssd1306_WriteString(Temperature1, Font_7x10, White);
        ssd1306_DrawCircle(116, 12, 2, White);  //To print degree only
        ssd1306_SetCursor (112,2);  //To print celcius
        ssd1306_WriteString("C", Font_11x18, White);
        ssd1306_UpdateScreen(); //display
        HAL_Delay(2000);
        ssd1306_SetCursor (0,24);
        ssd1306_WriteString("Pressure", Font_7x10, White);
        ssd1306_SetCursor(57,24);
        sprintf(Pressure1, "%.2f", Pressure);
        ssd1306_WriteString(Pressure1, Font_7x10, White);
        ssd1306_SetCursor(116,24);
        ssd1306_WriteString("pa", Font_7x10, White);
```

```
            ssd1306_UpdateScreen(); //display
            HAL_Delay(2000);
            ssd1306_SetCursor(0,36);
            ssd1306_WriteString("Altitude", Font_7x10, White);
            ssd1306_SetCursor (81,36);
            sprintf(Altitude1, "%.2f", Altitude);
            ssd1306_WriteString (Altitude1, Font_7x10, White);
            ssd1306_SetCursor(116,36);
            ssd1306_WriteString("m", Font_7x10, White);
            ssd1306_UpdateScreen(); //display
            HAL_Delay(2000);
  }
  /* USER CODE END 3 */
}
/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
  /** Configure the main internal regulator output voltage
  */
  __HAL_RCC_PWR_CLK_ENABLE();
  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
  /** Initializes the RCC Oscillators according to the specified parameters
  * in the RCC_OscInitTypeDef structure.
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
  RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
  RCC_OscInitStruct.PLL.PLLM = 4;
  RCC_OscInitStruct.PLL.PLLN = 180;
  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
  RCC_OscInitStruct.PLL.PLLQ = 4;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    Error_Handler();
  }
  /** Activate the Over-Drive mode
```

```
  */
  if (HAL_PWREx_EnableOverDrive() != HAL_OK)
  {
    Error_Handler();
  }
  /** Initializes the CPU, AHB and APB buses clocks
  */
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;
  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != HAL_OK)
  {
    Error_Handler();
  }
}
/**
  * @brief I2C1 Initialization Function
  * @param None
  * @retval None
  */
static void MX_I2C1_Init(void)
{
  /* USER CODE BEGIN I2C1_Init 0 */
  /* USER CODE END I2C1_Init 0 */
  /* USER CODE BEGIN I2C1_Init 1 */
  /* USER CODE END I2C1_Init 1 */
  hi2c1.Instance = I2C1;
  hi2c1.Init.ClockSpeed = 400000;
  hi2c1.Init.DutyCycle = I2C_DUTYCYCLE_2;
  hi2c1.Init.OwnAddress1 = 0;
  hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
  hi2c1.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
  hi2c1.Init.OwnAddress2 = 0;
  hi2c1.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
  hi2c1.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
  if (HAL_I2C_Init(&hi2c1) != HAL_OK)
  {
    Error_Handler();
  }
```

```
  /** Configure Analogue filter
  */
  if (HAL_I2CEx_ConfigAnalogFilter(&hi2c1, I2C_ANALOGFILTER_ENABLE) != HAL_OK)
  {
    Error_Handler();
  }
  /** Configure Digital filter
  */
  if (HAL_I2CEx_ConfigDigitalFilter(&hi2c1, 0) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN I2C1_Init 2 */

  /* USER CODE END I2C1_Init 2 */
}
/**
  * @brief GPIO Initialization Function
  * @param None
  * @retval None
  */
static void MX_GPIO_Init(void)
{

  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOH_CLK_ENABLE();
  __HAL_RCC_GPIOA_CLK_ENABLE();
  __HAL_RCC_GPIOB_CLK_ENABLE();
}
/* USER CODE BEGIN 4 */
/* USER CODE END 4 */
/**
  * @brief  This function is executed in case of error occurrence.
  * @retval None
  */
void Error_Handler(void)
{
  /* USER CODE BEGIN Error_Handler_Debug */
  /* User can add his own implementation to report the HAL error return state */
  __disable_irq();
  while (1)
  {
  }
```

```
  /* USER CODE END Error_Handler_Debug */
}
#ifdef  USE_FULL_ASSERT
/**
  * @brief  Reports the name of the source file and the source line number
  *         where the assert_param error has occurred.
  * @param  file: pointer to the source file name
  * @param  line: assert_param error line source number
  * @retval None
  */
void assert_failed(uint8_t *file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
  /* User can add his own implementation to report the file name and line number,
     ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
  /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */
```

BMP180 libraries

BMP180.c [39]

```
#include "stm32f4xx_hal.h"
#include "math.h"
extern I2C_HandleTypeDef hi2c1;
#define BMP180_I2C &hi2c1
#define BMP180_ADDRESS 0xEE


// Defines according to the datsheet
short AC1 = 0;
```

```
short AC2 = 0;
short AC3 = 0;
unsigned short AC4 = 0;
unsigned short AC5 = 0;
unsigned short AC6 = 0;
short B1 = 0;
short B2 = 0;
short MB = 0;
short MC = 0;
short MD = 0;
/******************/
long UT = 0;
short oss = 0;
long UP = 0;
long X1 = 0;
long X2 = 0;
long X3 = 0;
long B3 = 0;
long B5 = 0;
unsigned long B4 = 0;
long B6 = 0;
unsigned long B7 = 0;
/******************/
long Press = 0;
long Temp = 0;
#define atmPress 101325 //Pa
void read_calliberation_data (void)
{
	uint8_t Callib_Data[22] = {0};
	uint16_t Callib_Start = 0xAA;
	HAL_I2C_Mem_Read(BMP180_I2C, BMP180_ADDRESS, Callib_Start, 1, Callib_Data,22, HAL_MAX_DELAY);
	AC1 = ((Callib_Data[0] << 8) | Callib_Data[1]);
	AC2 = ((Callib_Data[2] << 8) | Callib_Data[3]);
	AC3 = ((Callib_Data[4] << 8) | Callib_Data[5]);
	AC4 = ((Callib_Data[6] << 8) | Callib_Data[7]);
	AC5 = ((Callib_Data[8] << 8) | Callib_Data[9]);
	AC6 = ((Callib_Data[10] << 8) | Callib_Data[11]);
	B1 = ((Callib_Data[12] << 8) | Callib_Data[13]);
	B2 = ((Callib_Data[14] << 8) | Callib_Data[15]);
	MB = ((Callib_Data[16] << 8) | Callib_Data[17]);
```

```
	MC = ((Callib_Data[18] << 8) | Callib_Data[19]);
	MD = ((Callib_Data[20] << 8) | Callib_Data[21]);

}
// Get uncompensated Temp
uint16_t Get_UTemp (void)
{
	uint8_t datatowrite = 0x2E;
	uint8_t Temp_RAW[2] = {0};
	HAL_I2C_Mem_Write(BMP180_I2C, BMP180_ADDRESS, 0xF4, 1, &datatowrite, 1, 1000);
	HAL_Delay (5);  // wait 4.5 ms
	HAL_I2C_Mem_Read(BMP180_I2C, BMP180_ADDRESS, 0xF6, 1, Temp_RAW, 2, 1000);
	return ((Temp_RAW[0]<<8) + Temp_RAW[1]);
}
float BMP180_GetTemp (void)
{
	UT = Get_UTemp();
	X1 = ((UT-AC6) * (AC5/(pow(2,15))));
	X2 = ((MC*(pow(2,11))) / (X1+MD));
	B5 = X1+X2;
	Temp = (B5+8)/(pow(2,4));
	return Temp/10.0;
}
// Get uncompensated Pressure
uint32_t Get_UPress (int oss)   // oversampling setting 0,1,2,3
{
	uint8_t datatowrite = 0x34+(oss<<6);
	uint8_t Press_RAW[3] = {0};
	HAL_I2C_Mem_Write(BMP180_I2C, BMP180_ADDRESS, 0xF4, 1, &datatowrite, 1, 1000);
	switch (oss)
	{
	  case (0):
			HAL_Delay (5);
			break;
	  case (1):
			HAL_Delay (8);
			break;
	  case (2):
			HAL_Delay (14);
			break;
```

```
            case (3):
                    HAL_Delay (26);
                    break;
            }
            HAL_I2C_Mem_Read(BMP180_I2C, BMP180_ADDRESS, 0xF6, 1, Press_RAW, 3, 1000);
            return (((Press_RAW[0]<<16)+(Press_RAW[1]<<8)+Press_RAW[2]) >> (8-oss));
}
float BMP180_GetPress (int oss)
{
            UP = Get_UPress(oss);
            X1 = ((UT-AC6) * (AC5/(pow(2,15))));
            X2 = ((MC*(pow(2,11))) / (X1+MD));
            B5 = X1+X2;
            B6 = B5-4000;
            X1 = (B2 * (B6*B6/(pow(2,12))))/(pow(2,11));
            X2 = AC2*B6/(pow(2,11));
            X3 = X1+X2;
            B3 = (((AC1*4+X3)<<oss)+2)/4;
            X1 = AC3*B6/pow(2,13);
            X2 = (B1 * (B6*B6/(pow(2,12))))/(pow(2,16));
            X3 = ((X1+X2)+2)/pow(2,2);
            B4 = AC4*(unsigned long)(X3+32768)/(pow(2,15));
            B7 = ((unsigned long)UP-B3)*(50000>>oss);
            if (B7<0x80000000) Press = (B7*2)/B4;
            else Press = (B7/B4)*2;
            X1 = (Press/(pow(2,8)))*(Press/(pow(2,8)));
            X1 = (X1*3038)/(pow(2,16));
            X2 = (-7357*Press)/(pow(2,16));
            Press = Press + (X1+X2+3791)/(pow(2,4));
            return Press;
}
float BMP180_GetAlt (int oss)
{
            BMP180_GetPress (oss);
            return 44330*(1-(pow((Press/(float)atmPress), 0.19029495718)));
}
void BMP180_Start (void)
{
            read_calliberation_data();
}
```

BMP180.h [39]

```
#ifndef _BMP180_H_
#define _BMP180_H_
#include "stm32f4xx_hal.h"
void BMP180_Start (void);
float BMP180_GetTemp (void);
float BMP180_GetPress (int oss);
float BMP180_GetAlt (int oss);
#endif /* INC_BMP180_H_ */
```

GPS NEO6 main

```
/* USER CODE END Header */
/* Includes ------------------------------------------------------------------*/
#include "main.h"
/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <stdbool.h>
#include <string.h>
#include <stddef.h>
```

```
#include <math.h>
/* USER CODE END Includes */
/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN PTD */
/* USER CODE END PTD */
/* Private define ------------------------------------------------------------*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */
/* Private macro -------------------------------------------------------------*/
/* USER CODE BEGIN PM */
/* USER CODE END PM */
/* Private variables ---------------------------------------------------------*/
UART_HandleTypeDef huart2;
DMA_HandleTypeDef hdma_usart2_rx;
DMA_HandleTypeDef hdma_usart2_tx;
/* USER CODE BEGIN PV */
/* USER CODE END PV */
/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
static void MX_USART2_UART_Init(void);
/* USER CODE BEGIN PFP */
/* USER CODE END PFP */
/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
int flag=0;

char  *nmea_longitude;
char  *nmea_latitude;
            char  *ns;
            char  *ew;
            char  *chk;
          // NMEA GPRMC
            char *utc_time;
            char *date;
            char *validity;
            char *speed_k;
            char *course_d;
            char *variation;
```

```
  char  *varia_dir;
 // Valori convertiti da gradi minuti secondi NMEA a gradi decimali
  float dec_longitude;
  float dec_latitude;
 // Paremetri di orario e calendario
  char    ora[2+1];
  char    minuto[2+1];
  char    secondo[2+1];
  char    giorno[2+1];
  char    mese[2+1];
  char    anno[2+1];
 //ora, minuto, secondo nella corretta forma hh:mm:ss
  char utc_corretto[8];
  char latitude_degree[2+1];
  char latitude_minute[10+1];
  char longitude_degree[3+1];
  char longitude_minute[10+1];
  //checksum_pointer per identificare la posizione dell'asterisco e il checksum number per salvare il numero relativo al checksum
  char *checksum_pointer;
  char checksum_number[3];
  uint8_t buffer_iniziale[255];
  char NMEA_MSG[80];
  uint8_t cnt=0;
  char msg_neo6[255];
void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
 flag=1;
 HAL_UART_Receive_DMA(&huart2,buffer_iniziale,255);
}
int nmea0183_checksum(char *msg){
 int checksum=0;
 int j=0;
 for(j=1;j<strlen(msg)-4;j++){
  checksum=checksum^(unsigned) msg[j];
 }
 return checksum;
}
void GPS_Extractor(char* NMEA_MSG)
{
```

```
        if(flag==1){
        memset(msg_neo6,0,255);
        sprintf(msg_neo6,"%s",buffer_iniziale);
        char *token,*string;
        string=strdup(msg_neo6);
        //separiamo le stringe una dall'altra per poterle analizzare singolarmente
        while((token=strsep(&string,"\n"))!=NULL){
   memset (NMEA_MSG,0,80);
        sprintf(NMEA_MSG,"%s",token);



        //Controllo per assicurarsi che la stringa sia di tipo GPRMC e della presenza dell'asterisco relativo al checksum
        if ((strstr(NMEA_MSG, "$GPRMC") != 0) && strstr(NMEA_MSG, "*") != 0){
          //cercare nel messaggio GPRMC ricevuto l'asterico
          checksum_pointer = strstr(NMEA_MSG, "*");
          //attraverso il puntatore inserire in checksum_number i 2 valori relativi al checksum
          memcpy(checksum_number, &checksum_pointer[1], 2);
          checksum_number[2] = '\0';
          uint8_t intSum = nmea0183_checksum(NMEA_MSG);
          char verifica[2];
          //All'interno di verifica scriviamo intSum che sorrisponde al checksum
          sprintf(verifica,"%X",intSum);
          //Dobbiamo verificare che all'interno di cheksum_number
          //ci sia "verifica", cioè che i due corrispondano per capire se
          //effettivamente il messaggio che abbiamo acquisito sia corretto
          if(strstr(checksum_number,verifica)!=NULL){
                  cnt=0;
                  //separiamo il messaggio acquisito in base alla posizione della virgola che delimita le diverse sezioni
                  //e acquisiamo i dati implementando una logica switch-case secondo l'ordine in cui il messaggio di tipo
GPRMC ce li fornisce

                  for(char *pV=strtok(NMEA_MSG,",");pV!=NULL;pV=strtok(NULL,",")){
                          switch(cnt){
                          case 1:
                                  utc_time=strdup(pV);
                                  break;
                          case 2:
                                  validity=strdup(pV);
                                  break;
                          case 3:
```

```
                    nmea_latitude=strdup(pV);
                    break;
                case 4:
                    ns=strdup(pV);
                    break;
                case 5:
                    nmea_longitude=strdup(pV);
                    break;
                case 6:
                    ew=strdup(pV);
                    break;
                case 7:
                    speed_k=strdup(pV);
                    break;
                case 8:
                    course_d=strdup(pV);
                    break;
                case 9:
                    date=strdup(pV);
                    break;
                case 10:
                    variation=strdup(pV);
                    break;
                case 11:
                    varia_dir=strdup(pV);
                    break;
                default:
                    break;
                }

cnt++;
            }
            memcpy(latitude_degree, &nmea_latitude[0], 2);
            latitude_degree[2] = '\0';
            memcpy(latitude_minute, &nmea_latitude[2], 10);
            latitude_minute[10] = '\0';
            memcpy(longitude_degree, &nmea_longitude[0], 3);
            longitude_degree[3] = '\0';
            memcpy(longitude_minute, &nmea_longitude[3], 10);
            longitude_minute[10] = '\0';
```

```
                    //nella funzione seguente si prendono i valori da utc_time e si dividono in ora, minuto e secondo
                    memcpy(ora, &utc_time[0], 2);
                    ora[2] = '\0';
                    memcpy(minuto, &utc_time[2], 2);
                    minuto[2] = '\0';
                    memcpy(secondo, &utc_time[4], 2);
                    secondo[2] = '\0';
   }
  }
 }
        flag=0;
}
}
/* USER CODE END 0 */
/**
  * @brief  The application entry point.
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */
  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/
  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();
  /* USER CODE BEGIN Init */
  /* USER CODE END Init */
  /* Configure the system clock */
  SystemClock_Config();
  /* USER CODE BEGIN SysInit */
  /* USER CODE END SysInit */
  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_DMA_Init();
  MX_USART2_UART_Init();
  /* USER CODE BEGIN 2 */
  HAL_UART_Receive_DMA(&huart2,(uint8_t*)buffer_iniziale,255);
  /* USER CODE END 2 */
  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
```

```
  while (1)
  {
            GPS_Extractor(NMEA_MSG);
    /* USER CODE END WHILE */
    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
}
/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

  /** Configure the main internal regulator output voltage
  */
  __HAL_RCC_PWR_CLK_ENABLE();
  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
  /** Initializes the RCC Oscillators according to the specified parameters
  * in the RCC_OscInitTypeDef structure.
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
  RCC_OscInitStruct.HSIState = RCC_HSI_ON;
  RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
  RCC_OscInitStruct.PLL.PLLM = 8;
  RCC_OscInitStruct.PLL.PLLN = 180;
  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
  RCC_OscInitStruct.PLL.PLLQ = 4;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    Error_Handler();
  }
  /** Activate the Over-Drive mode
  */
  if (HAL_PWREx_EnableOverDrive() != HAL_OK)
  {
    Error_Handler();
```

```
  }
  /** Initializes the CPU, AHB and APB buses clocks
  */
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;
  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != HAL_OK)
  {
    Error_Handler();
  }
}
/**
  * @brief USART2 Initialization Function
  * @param None
  * @retval None
  */
static void MX_USART2_UART_Init(void)
{
  /* USER CODE BEGIN USART2_Init 0 */
  /* USER CODE END USART2_Init 0 */
  /* USER CODE BEGIN USART2_Init 1 */
  /* USER CODE END USART2_Init 1 */
  huart2.Instance = USART2;
  huart2.Init.BaudRate = 9600;
  huart2.Init.WordLength = UART_WORDLENGTH_8B;
  huart2.Init.StopBits = UART_STOPBITS_1;
  huart2.Init.Parity = UART_PARITY_NONE;
  huart2.Init.Mode = UART_MODE_TX_RX;
  huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
  huart2.Init.OverSampling = UART_OVERSAMPLING_16;
  if (HAL_UART_Init(&huart2) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN USART2_Init 2 */
  /* USER CODE END USART2_Init 2 */
}
/**
  * Enable DMA controller clock
```

```
  */
static void MX_DMA_Init(void)
{
  /* DMA controller clock enable */
  __HAL_RCC_DMA1_CLK_ENABLE();

  /* DMA interrupt init */
  /* DMA1_Stream5_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(DMA1_Stream5_IRQn, 0, 0);
  HAL_NVIC_EnableIRQ(DMA1_Stream5_IRQn);
  /* DMA1_Stream6_IRQn interrupt configuration */
  HAL_NVIC_SetPriority(DMA1_Stream6_IRQn, 0, 0);
  HAL_NVIC_EnableIRQ(DMA1_Stream6_IRQn);

}
/**
  * @brief GPIO Initialization Function
  * @param None
  * @retval None
  */
static void MX_GPIO_Init(void)
{
  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOH_CLK_ENABLE();
  __HAL_RCC_GPIOA_CLK_ENABLE();
  __HAL_RCC_GPIOD_CLK_ENABLE();

}
/* USER CODE BEGIN 4 */

/* USER CODE END 4 */

/**
  * @brief  This function is executed in case of error occurrence.
  * @retval None
  */
void Error_Handler(void)
{
  /* USER CODE BEGIN Error_Handler_Debug */
  /* User can add his own implementation to report the HAL error return state */
  __disable_irq();
  while (1)
  {
```

```
  }
  /* USER CODE END Error_Handler_Debug */
}
#ifdef  USE_FULL_ASSERT
/**
  * @brief  Reports the name of the source file and the source line number
  *         where the assert_param error has occurred.
  * @param  file: pointer to the source file name
  * @param  line: assert_param error line source number
  * @retval None
  */
void assert_failed(uint8_t *file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
  /* User can add his own implementation to report the file name and line number,
     ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
  /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */
```

SDCARD main

```
/* USER CODE END Header */
/* Includes ------------------------------------------------------------------*/
#include "main.h"
#include "fatfs.h"
/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
#include "fatfs_sd.h"
#include "string.h"
#include "BMP180.h"
/* USER CODE END Includes */
/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN PTD */
/* USER CODE END PTD */
/* Private define ------------------------------------------------------------*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */
/* Private macro -------------------------------------------------------------*/
/* USER CODE BEGIN PM */
/* USER CODE END PM */
/* Private variables ---------------------------------------------------------*/
I2C_HandleTypeDef hi2c1;
SPI_HandleTypeDef hspi1;
UART_HandleTypeDef huart1;
/* USER CODE BEGIN PV */
/* USER CODE END PV */
/* Private function prototypes -----------------------------------------------*/
```

```
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_SPI1_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_I2C1_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
float Temperature;
float Pressure;
float Altitude;
char Temperature1[10];
char Pressure1[10];
char Altitude1[10];
FATFS fs;  // file system
FIL fil; // File
FILINFO fno;
FRESULT fresult;  // result
UINT br, bw;  // File read/write count
/**** capacity related ****/
FATFS *pfs;
DWORD fre_clust;
uint32_t total, free_space;
#define BUFFER_SIZE 128
char buffer[BUFFER_SIZE];  // to store strings..
int i=0;
int bufsize (char *buf)
{
	int i=0;
	while (*buf++ != '\0') i++;
	return i;
}
void clear_buffer (void)
{
	for (int i=0; i<BUFFER_SIZE; i++) buffer[i] = '\0';
}
void send_uart (char *string)
```

```
{
	uint8_t len = strlen (string);
	HAL_UART_Transmit(&huart1, (uint8_t *) string, len, HAL_MAX_DELAY);  // transmit in blocking mode
}
```

```
/* USER CODE END 0 */
/**
  * @brief  The application entry point.
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */
  /* USER CODE END 1 */
  /* MCU Configuration--------------------------------------------------------*/
  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();
  /* USER CODE BEGIN Init */
  /* USER CODE END Init */
  /* Configure the system clock */
  SystemClock_Config();
  /* USER CODE BEGIN SysInit */
  /* USER CODE END SysInit */
  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_SPI1_Init();
  MX_USART1_UART_Init();
  MX_FATFS_Init();
  MX_I2C1_Init();
  /* USER CODE BEGIN 2 */
  BMP180_Start();
  Temperature=BMP180_GetTemp();
  Pressure=BMP180_GetPress(0);
  Altitude=BMP180_GetAlt(0);
  HAL_Delay (500);
  fresult = f_mount(&fs, "/", 1);
```

```
        if (fresult != FR_OK) send_uart ("ERROR!!! in mounting SD CARD...\n\n");
        else send_uart("SD CARD mounted successfully...\n\n");
        /********** Card capacity details ********************/
        /* Check free space */
        f_getfree("", &fre_clust, &pfs);
        total = (uint32_t)((pfs->n_fatent - 2) * pfs->csize * 0.5);
        sprintf (buffer, "SD CARD Total Size: \t%lu\n",total);
        send_uart(buffer);
        clear_buffer();
        free_space = (uint32_t)(fre_clust * pfs->csize * 0.5);
        sprintf (buffer, "SD CARD Free Space: \t%lu\n\n",free_space);
        send_uart(buffer);
        clear_buffer();
        /************ The following operation is using PUTS and GETS *********************/
        /* Open file to write/ create a file if it doesn't exist */
    fresult = f_open(&fil, "file1.txt", FA_OPEN_ALWAYS | FA_READ | FA_WRITE);
        /* Writing text */
  printf(buffer, " Temperature= %f C \n Pressure=%f Pa \n     Altitude=%f m \n",Temperature,Pressure,Altitude);
  fresult=f_puts(buffer,&fil);
        /* Close file */
        fresult = f_close(&fil);
        if (fresult == FR_OK)send_uart ("File1.txt created and the data is written \n");
        /* Open file to read */
        fresult = f_open(&fil, "file1.txt", FA_READ);
        /* Read string from the file */
        f_gets(buffer, f_size(&fil), &fil);
        send_uart("File1.txt is opened and it contains the data as shown below\n");
        send_uart(buffer);
        send_uart("\n\n");
        /* Close file */
        f_close(&fil);
        clear_buffer();
/* USER CODE END 2 */
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
```

```
  }
  /* USER CODE END 3 */
}
/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

  /** Configure the main internal regulator output voltage
  */
  __HAL_RCC_PWR_CLK_ENABLE();
  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE3);
  /** Initializes the RCC Oscillators according to the specified parameters
  * in the RCC_OscInitTypeDef structure.
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
  RCC_OscInitStruct.HSEState = RCC_HSE_ON;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
  RCC_OscInitStruct.PLL.PLLM = 4;
  RCC_OscInitStruct.PLL.PLLN = 72;
  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
  RCC_OscInitStruct.PLL.PLLQ = 4;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    Error_Handler();
  }
  /** Initializes the CPU, AHB and APB buses clocks
  */
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                              |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
  {
```

```
    Error_Handler();
  }
}
/**
  * @brief I2C1 Initialization Function
  * @param None
  * @retval None
  */
static void MX_I2C1_Init(void)
{
  /* USER CODE BEGIN I2C1_Init 0 */
  /* USER CODE END I2C1_Init 0 */
  /* USER CODE BEGIN I2C1_Init 1 */
  /* USER CODE END I2C1_Init 1 */
  hi2c1.Instance = I2C1;
  hi2c1.Init.ClockSpeed = 100000;
  hi2c1.Init.DutyCycle = I2C_DUTYCYCLE_2;
  hi2c1.Init.OwnAddress1 = 0;
  hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
  hi2c1.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
  hi2c1.Init.OwnAddress2 = 0;
  hi2c1.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
  hi2c1.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
  if (HAL_I2C_Init(&hi2c1) != HAL_OK)
  {
    Error_Handler();
  }
  /** Configure Analogue filter
  */
  if (HAL_I2CEx_ConfigAnalogFilter(&hi2c1, I2C_ANALOGFILTER_ENABLE) != HAL_OK)
  {
    Error_Handler();
  }
  /** Configure Digital filter
  */
  if (HAL_I2CEx_ConfigDigitalFilter(&hi2c1, 0) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN I2C1_Init 2 */
  /* USER CODE END I2C1_Init 2 */
}
```

```
/**
  * @brief SPI1 Initialization Function
  * @param None
  * @retval None
  */
static void MX_SPI1_Init(void)
{
  /* USER CODE BEGIN SPI1_Init 0 */
  /* USER CODE END SPI1_Init 0 */
  /* USER CODE BEGIN SPI1_Init 1 */
  /* USER CODE END SPI1_Init 1 */
  /* SPI1 parameter configuration*/
  hspi1.Instance = SPI1;
  hspi1.Init.Mode = SPI_MODE_MASTER;
  hspi1.Init.Direction = SPI_DIRECTION_2LINES;
  hspi1.Init.DataSize = SPI_DATASIZE_8BIT;
  hspi1.Init.CLKPolarity = SPI_POLARITY_LOW;
  hspi1.Init.CLKPhase = SPI_PHASE_1EDGE;
  hspi1.Init.NSS = SPI_NSS_SOFT;
  hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_4;
  hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
  hspi1.Init.TIMode = SPI_TIMODE_DISABLE;
  hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
  hspi1.Init.CRCPolynomial = 10;
  if (HAL_SPI_Init(&hspi1) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN SPI1_Init 2 */
  /* USER CODE END SPI1_Init 2 */
}

/**
  * @brief USART1 Initialization Function
  * @param None
  * @retval None
  */
static void MX_USART1_UART_Init(void)
{
  /* USER CODE BEGIN USART1_Init 0 */
  /* USER CODE END USART1_Init 0 */
  /* USER CODE BEGIN USART1_Init 1 */
```

```
  /* USER CODE END USART1_Init 1 */
  huart1.Instance = USART1;
  huart1.Init.BaudRate = 115200;
  huart1.Init.WordLength = UART_WORDLENGTH_8B;
  huart1.Init.StopBits = UART_STOPBITS_1;
  huart1.Init.Parity = UART_PARITY_NONE;
  huart1.Init.Mode = UART_MODE_TX_RX;
  huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
  huart1.Init.OverSampling = UART_OVERSAMPLING_16;
  if (HAL_UART_Init(&huart1) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN USART1_Init 2 */

  /* USER CODE END USART1_Init 2 */
}
/**
  * @brief GPIO Initialization Function
  * @param None
  * @retval None
  */
static void MX_GPIO_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};
  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOH_CLK_ENABLE();
  __HAL_RCC_GPIOA_CLK_ENABLE();
  __HAL_RCC_GPIOB_CLK_ENABLE();
  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, GPIO_PIN_RESET);
  /*Configure GPIO pin : PB6 */
  GPIO_InitStruct.Pin = GPIO_PIN_6;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

}
/* USER CODE BEGIN 4 */
/* USER CODE END 4 */
/**
```

```
  * @brief  This function is executed in case of error occurrence.
  * @retval None
  */
void Error_Handler(void)
{
  /* USER CODE BEGIN Error_Handler_Debug */
  /* User can add his own implementation to report the HAL error return state */
  __disable_irq();
  while (1)
  {
  }
  /* USER CODE END Error_Handler_Debug */
}
#ifdef  USE_FULL_ASSERT
/**
  * @brief  Reports the name of the source file and the source line number
  *         where the assert_param error has occurred.
  * @param  file: pointer to the source file name
  * @param  line: assert_param error line source number
  * @retval None
  */
void assert_failed(uint8_t *file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
  /* User can add his own implementation to report the file name and line number,
     ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
  /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */
```

SDCARD libraries

fatfs_sd.c [39]

```
#define TRUE  1
#define FALSE 0
#define bool BYTE
#include "stm32f4xx_hal.h"
#include "diskio.h"
#include "fatfs_sd.h"
extern SPI_HandleTypeDef        hspi1;
#define HSPI_SDCARD                     &hspi1
#defineSD_CS_PORT                       GPIOB
#define SD_CS_PIN                       GPIO_PIN_6
extern volatile uint16_t Timer1, Timer2;                                        /* 1ms Timer Counter */
static volatile DSTATUS Stat = STA_NOINIT;          /* Disk Status */
static uint8_t CardType;                /* Type 0:MMC, 1:SDC, 2:Block addressing */
```

```
static uint8_t PowerFlag = 0;                                    /* Power flag */
/***************************************
 * SPI functions
 ***************************************/
/* slave select */
static void SELECT(void)
{
        HAL_GPIO_WritePin(SD_CS_PORT, SD_CS_PIN, GPIO_PIN_RESET);
        HAL_Delay(1);
}
/* slave deselect */
static void DESELECT(void)
{
        HAL_GPIO_WritePin(SD_CS_PORT, SD_CS_PIN, GPIO_PIN_SET);
        HAL_Delay(1);
}
/* SPI transmit a byte */
static void SPI_TxByte(uint8_t data)
{
        while(!__HAL_SPI_GET_FLAG(HSPI_SDCARD, SPI_FLAG_TXE));
        HAL_SPI_Transmit(HSPI_SDCARD, &data, 1, SPI_TIMEOUT);
}
/* SPI transmit buffer */
static void SPI_TxBuffer(uint8_t *buffer, uint16_t len)
{
        while(!__HAL_SPI_GET_FLAG(HSPI_SDCARD, SPI_FLAG_TXE));
        HAL_SPI_Transmit(HSPI_SDCARD, buffer, len, SPI_TIMEOUT);
}

/* SPI receive a byte */
static uint8_t SPI_RxByte(void)
{
        uint8_t dummy, data;
        dummy = 0xFF;
        while(!__HAL_SPI_GET_FLAG(HSPI_SDCARD, SPI_FLAG_TXE));
        HAL_SPI_TransmitReceive(HSPI_SDCARD, &dummy, &data, 1, SPI_TIMEOUT);
        return data;
}
/* SPI receive a byte via pointer */
static void SPI_RxBytePtr(uint8_t *buff)
```

```
{
        *buff = SPI_RxByte();
}
/*************************************
 * SD functions
 *************************************/
/* wait SD ready */
static uint8_t SD_ReadyWait(void)
{
        uint8_t res;
        /* timeout 500ms */
        Timer2 = 500;
        /* if SD goes ready, receives 0xFF */
        do {
          res = SPI_RxByte();
        } while ((res != 0xFF) && Timer2);
        return res;
}
/* power on */
static void SD_PowerOn(void)
{
        uint8_t args[6];
        uint32_t cnt = 0x1FFF;

        /* transmit bytes to wake up */
        DESELECT();
        for(int i = 0; i < 10; i++)
        {
          SPI_TxByte(0xFF);
        }

        /* slave select */
        SELECT();
        /* make idle state */
        args[0] = CMD0;                 /* CMD0:GO_IDLE_STATE */
        args[1] = 0;
        args[2] = 0;
        args[3] = 0;
        args[4] = 0;
        args[5] = 0x95;                 /* CRC */
```

```
        SPI_TxBuffer(args, sizeof(args));
        /* wait response */
        while ((SPI_RxByte() != 0x01) && cnt)
        {
          cnt--;
        }
        DESELECT();
        SPI_TxByte(0XFF);
        PowerFlag = 1;
}
/* power off */
static void SD_PowerOff(void)
{
        PowerFlag = 0;
}
/* check power flag */
static uint8_t SD_CheckPower(void)
{
        return PowerFlag;
}
/* receive data block */
static bool SD_RxDataBlock(BYTE *buff, UINT len)
{
        uint8_t token;
        /* timeout 200ms */
        Timer1 = 200;
        /* loop until receive a response or timeout */
        do {
          token = SPI_RxByte();
        } while((token == 0xFF) && Timer1);

        /* invalid response */
        if(token != 0xFE) return FALSE;
        /* receive data */
        do {
          SPI_RxBytePtr(buff++);
        } while(len--);
        /* discard CRC */
        SPI_RxByte();
        SPI_RxByte();
```

```
	return TRUE;
}
/* transmit data block */
#if _USE_WRITE == 1
static bool SD_TxDataBlock(const uint8_t *buff, BYTE token)
{
	uint8_t resp;
	uint8_t i = 0;
	/* wait SD ready */
	if (SD_ReadyWait() != 0xFF) return FALSE;
	/* transmit token */
	SPI_TxByte(token);
	/* if it's not STOP token, transmit data */
	if (token != 0xFD)
	{
	  SPI_TxBuffer((uint8_t*)buff, 512);
	  /* discard CRC */
	  SPI_RxByte();
	  SPI_RxByte();
	  /* receive response */
	  while (i <= 64)
	  {
			resp = SPI_RxByte();
			/* transmit 0x05 accepted */
			if ((resp & 0x1F) == 0x05) break;
			i++;
	  }
	  /* recv buffer clear */
	  while (SPI_RxByte() == 0);
	}

	/* transmit 0x05 accepted */
	if ((resp & 0x1F) == 0x05) return TRUE;
	return FALSE;
}
#endif /* _USE_WRITE */
/* transmit command */
static BYTE SD_SendCmd(BYTE cmd, uint32_t arg)
{
	uint8_t crc, res;
```

```
	/* wait SD ready */
	if (SD_ReadyWait() != 0xFF) return 0xFF;
	/* transmit command */
	SPI_TxByte(cmd);                                    /* Command */
	SPI_TxByte((uint8_t)(arg >> 24));     /* Argument[31..24] */
	SPI_TxByte((uint8_t)(arg >> 16));     /* Argument[23..16] */
	SPI_TxByte((uint8_t)(arg >> 8));      /* Argument[15..8] */
	SPI_TxByte((uint8_t)arg);                   /* Argument[7..0] */
	/* prepare CRC */
	if(cmd == CMD0) crc = 0x95;           /* CRC for CMD0(0) */
	else if(cmd == CMD8) crc = 0x87;      /* CRC for CMD8(0x1AA) */
	else crc = 1;
	/* transmit CRC */
	SPI_TxByte(crc);

	/* Skip a stuff byte when STOP_TRANSMISSION */
	if (cmd == CMD12) SPI_RxByte();

	/* receive response */
	uint8_t n = 10;
	do {
	  res = SPI_RxByte();
	} while ((res & 0x80) && --n);
	return res;
}

/*-----------------------------------
 * user_diskio.c functions
 -----------------------------------*/

/* initialize SD */
DSTATUS SD_disk_initialize(BYTE drv)
{
	uint8_t n, type, ocr[4];

	/* single drive, drv should be 0 */
	if(drv) return STA_NOINIT;
	/* no disk */
	if(Stat & STA_NODISK) return Stat;
	/* power on */
```

```
	SD_PowerOn();
	/* slave select */
	SELECT();
	/* check disk type */
	type = 0;
	/* send GO_IDLE_STATE command */
	if (SD_SendCmd(CMD0, 0) == 1)
	{
	  /* timeout 1 sec */
	  Timer1 = 1000;

	  /* SDC V2+ accept CMD8 command, http://elm-chan.org/docs/mmc/mmc_e.html */
	  if (SD_SendCmd(CMD8, 0x1AA) == 1)
	  {
		/* operation condition register */
		for (n = 0; n < 4; n++)
		{
			ocr[n] = SPI_RxByte();
		}

		/* voltage range 2.7-3.6V */
		if (ocr[2] == 0x01 && ocr[3] == 0xAA)
		{
			/* ACMD41 with HCS bit */
			do {
				if (SD_SendCmd(CMD55, 0) <= 1 && SD_SendCmd(CMD41, 1UL << 30) == 0)
break;
			} while (Timer1);
			/* READ_OCR */
			if (Timer1 && SD_SendCmd(CMD58, 0) == 0)
			{
				/* Check CCS bit */
				for (n = 0; n < 4; n++)
				{
					ocr[n] = SPI_RxByte();
				}

				/* SDv2 (HC or SC) */
				type = (ocr[0] & 0x40) ? CT_SD2 | CT_BLOCK : CT_SD2;
			}
```

```
                }
        }
        else
        {
                /* SDC V1 or MMC */
                type = (SD_SendCmd(CMD55, 0) <= 1 && SD_SendCmd(CMD41, 0) <= 1) ? CT_SD1 : CT_MMC;

                do
                {
                        if (type == CT_SD1)
                        {
                                if (SD_SendCmd(CMD55, 0) <= 1 && SD_SendCmd(CMD41, 0) == 0) break; /*
ACMD41 */
                        }
                        else
                        {
                                if (SD_SendCmd(CMD1, 0) == 0) break; /* CMD1 */
                        }

                } while (Timer1);

                /* SET_BLOCKLEN */
                if (!Timer1 || SD_SendCmd(CMD16, 512) != 0) type = 0;
        }
    }
    CardType = type;
    /* Idle */
    DESELECT();
    SPI_RxByte();
    /* Clear STA_NOINIT */
    if (type)
    {
      Stat &= ~STA_NOINIT;
    }
    else
    {
      /* Initialization failed */
      SD_PowerOff();
    }
    return Stat;
```

```
}
/* return disk status */
DSTATUS SD_disk_status(BYTE drv)
{
	if (drv) return STA_NOINIT;
	return Stat;
}
/* read sector */
DRESULT SD_disk_read(BYTE pdrv, BYTE* buff, DWORD sector, UINT count)
{
	/* pdrv should be 0 */
	if (pdrv || !count) return RES_PARERR;
	/* no disk */
	if (Stat & STA_NOINIT) return RES_NOTRDY;
	/* convert to byte address */
	if (!(CardType & CT_SD2)) sector *= 512;
	SELECT();
	if (count == 1)
	{
	  /* READ_SINGLE_BLOCK */
	  if ((SD_SendCmd(CMD17, sector) == 0) && SD_RxDataBlock(buff, 512)) count = 0;
	}
	else
	{
	  /* READ_MULTIPLE_BLOCK */
	  if (SD_SendCmd(CMD18, sector) == 0)
	  {
		do {
			if (!SD_RxDataBlock(buff, 512)) break;
			buff += 512;
		} while (--count);

		/* STOP_TRANSMISSION */
		SD_SendCmd(CMD12, 0);
	  }
	}
	/* Idle */
	DESELECT();
	SPI_RxByte();
```

```
	return count ? RES_ERROR : RES_OK;
}
/* write sector */
#if _USE_WRITE == 1
DRESULT SD_disk_write(BYTE pdrv, const BYTE* buff, DWORD sector, UINT count)
{
	/* pdrv should be 0 */
	if (pdrv || !count) return RES_PARERR;

	/* no disk */
	if (Stat & STA_NOINIT) return RES_NOTRDY;

	/* write protection */
	if (Stat & STA_PROTECT) return RES_WRPRT;

	/* convert to byte address */
	if (!(CardType & CT_SD2)) sector *= 512;

	SELECT();

	if (count == 1)
	{
	  /* WRITE_BLOCK */
	  if ((SD_SendCmd(CMD24, sector) == 0) && SD_TxDataBlock(buff, 0xFE))
		count = 0;
	}
	else
	{
	  /* WRITE_MULTIPLE_BLOCK */
	  if (CardType & CT_SD1)
	  {
		SD_SendCmd(CMD55, 0);
		SD_SendCmd(CMD23, count); /* ACMD23 */
	  }

	  if (SD_SendCmd(CMD25, sector) == 0)
	  {
		do {
			if(!SD_TxDataBlock(buff, 0xFC)) break;
			buff += 512;
```

```
                } while (--count);

                /* STOP_TRAN token */
                if(!SD_TxDataBlock(0, 0xFD))
                {
                        count = 1;
                }
          }
        }
        /* Idle */
        DESELECT();
        SPI_RxByte();

        return count ? RES_ERROR : RES_OK;
}
#endif /* _USE_WRITE */

/* ioctl */
DRESULT SD_disk_ioctl(BYTE drv, BYTE ctrl, void *buff)
{
        DRESULT res;
        uint8_t n, csd[16], *ptr = buff;
        WORD csize;

        /* pdrv should be 0 */
        if (drv) return RES_PARERR;
        res = RES_ERROR;

        if (ctrl == CTRL_POWER)
        {
          switch (*ptr)
          {
          case 0:
                SD_PowerOff();          /* Power Off */
                res = RES_OK;
                break;
          case 1:
                SD_PowerOn();           /* Power On */
                res = RES_OK;
                break;
```

```
  case 2:
        *(ptr + 1) = SD_CheckPower();
        res = RES_OK;           /* Power Check */
        break;
  default:
        res = RES_PARERR;
  }
}
else
{
  /* no disk */
  if (Stat & STA_NOINIT) return RES_NOTRDY;
  SELECT();
  switch (ctrl)
  {
  case GET_SECTOR_COUNT:
        /* SEND_CSD */
        if ((SD_SendCmd(CMD9, 0) == 0) && SD_RxDataBlock(csd, 16))
        {
              if ((csd[0] >> 6) == 1)
              {
                    /* SDC V2 */
                    csize = csd[9] + ((WORD) csd[8] << 8) + 1;
                    *(DWORD*) buff = (DWORD) csize << 10;
              }
              else
              {
                    /* MMC or SDC V1 */
                    n = (csd[5] & 15) + ((csd[10] & 128) >> 7) + ((csd[9] & 3) << 1) + 2;
                    csize = (csd[8] >> 6) + ((WORD) csd[7] << 2) + ((WORD) (csd[6] & 3) << 10) + 1;
                    *(DWORD*) buff = (DWORD) csize << (n - 9);
              }
              res = RES_OK;
        }
        break;
  case GET_SECTOR_SIZE:
        *(WORD*) buff = 512;
        res = RES_OK;
        break;
  case CTRL_SYNC:
```

```
                if (SD_ReadyWait() == 0xFF) res = RES_OK;
                break;
        case MMC_GET_CSD:
                /* SEND_CSD */
                if (SD_SendCmd(CMD9, 0) == 0 && SD_RxDataBlock(ptr, 16)) res = RES_OK;
                break;
        case MMC_GET_CID:
                /* SEND_CID */
                if (SD_SendCmd(CMD10, 0) == 0 && SD_RxDataBlock(ptr, 16)) res = RES_OK;
                break;
        case MMC_GET_OCR:
                /* READ_OCR */
                if (SD_SendCmd(CMD58, 0) == 0)
                {
                        for (n = 0; n < 4; n++)
                        {
                                *ptr++ = SPI_RxByte();
                        }
                        res = RES_OK;
                }
        default:
                res = RES_PARERR;
        }
        DESELECT();
        SPI_RxByte();
    }
    return res;
}
```

fatfs_sd.h [39]

```
#ifndef __FATFS_SD_H
#define __FATFS_SD_H
/* Definitions for MMC/SDC command */
#define CMD0    (0x40+0)        /* GO_IDLE_STATE */
#define CMD1    (0x40+1)        /* SEND_OP_COND */
#define CMD8    (0x40+8)        /* SEND_IF_COND */
#define CMD9    (0x40+9)        /* SEND_CSD */
#define CMD10   (0x40+10)       /* SEND_CID */
```

```
#define CMD12   (0x40+12)       /* STOP_TRANSMISSION */
#define CMD16   (0x40+16)       /* SET_BLOCKLEN */
#define CMD17   (0x40+17)       /* READ_SINGLE_BLOCK */
#define CMD18   (0x40+18)       /* READ_MULTIPLE_BLOCK */
#define CMD23   (0x40+23)       /* SET_BLOCK_COUNT */
#define CMD24   (0x40+24)       /* WRITE_BLOCK */
#define CMD25   (0x40+25)       /* WRITE_MULTIPLE_BLOCK */
#define CMD41   (0x40+41)       /* SEND_OP_COND (ACMD) */
#define CMD55   (0x40+55)       /* APP_CMD */
#define CMD58   (0x40+58)       /* READ_OCR */
/* MMC card type flags (MMC_GET_TYPE) */
#define CT_MMC          0x01            /* MMC ver 3 */
#define CT_SD1          0x02            /* SD ver 1 */
#define CT_SD2          0x04            /* SD ver 2 */
#define CT_SDC          0x06            /* SD */
#define CT_BLOCK        0x08            /* Block addressing */
/* Functions */
DSTATUS SD_disk_initialize (BYTE pdrv);
DSTATUS SD_disk_status (BYTE pdrv);
DRESULT SD_disk_read (BYTE pdrv, BYTE* buff, DWORD sector, UINT count);
DRESULT SD_disk_write (BYTE pdrv, const BYTE* buff, DWORD sector, UINT count);
DRESULT SD_disk_ioctl (BYTE pdrv, BYTE cmd, void* buff);
#define SPI_TIMEOUT 100
#endif
```