TABLE OF CONTENTS

INTRODUCTION

With the presence of the Internet and its significant role in our life, everyone seeks to use it in all aspects of life to save time and money. There have been many types of research about linking anything surrounding us or the ecosystem to Internet. Also, there is the possibility to remote control things. The Internet of things has come because of these needs. This technology is based on connecting everything to the internet and remotely controlled either by phone applications, websites or even through cloud services.

In transportation, many people suffer from frequent problems e.g. traffic congestion, whether through cars or buses for their stuff that may cause a waste of time and money. In addition, the risk of accidents that are considered the most serious threat to people, as well as the various damage that may occur. For instance, if someone wants to take his son to his school or somewhere else, but cannot do that because of his business. In this case, driving his car will waste his time and make him lose a lot of money. Another example, in emergencies, it may be difficult to people to access some places or it may be done like very dangerous. In such cases, introducing the Internet of things using remote-operated ground vehicle can replace the man's presence obligation and, saves his time and money. Moreover, the heavy industry depends on many machines that are characterized by speed and accuracy in manufacturing, so it exceeds the human abilities. It contributes significantly to save time, effort and reduce the danger. But what about the way to control these machines? Is it possible to use ground vehicles remotely and without need the driver presence?

One of the promising solutions that researchers worked on is to invent powerful, performant and intelligent remote-operated unmanned vehicles. Hence, we get inspired by some literature projects that helped in producing this work.

This dissertation deals with the building of small remote-operated ground vehicles based on cheap and frequent available components, as well as it explains the necessary steps should be followed to obtain successful results. It will be done by passing through various stages, starting with planning the project's model including the vehicle and commands architecture, until performing experimental tests and examinations.

The intended vehicle will be composed of cheap and available components e.g. Arduino hardware, and ultrasonic and motor driver shield modules, that ensures the movement of the vehicle towards the four directions i.e. right, left, forward and back, according to Internet-based requests.

In addition, the system responsible for processing and sending movement orders will be present the server side which in turn will receive these requests from the Android-based mobile application.

1.1 motivation

Earlier, many types of industries were exclusive to the developed countries due to the difficulty and complexity of their manufacturing. These problems come from that products are, initially, manufactured from raw components that will, consequently, lead to the necessity to highly advanced machines. In fact, these machines will be considered as a secret arm for the developed countries as well as governments that own this type of machines will strive to keep other countries in permanent need of their products.

The aim of this work is to encourage and strengthen some type of industries in Algeria e.g. security, exploring and self-driving mobile vehicles. As it will demonstrate how from cheap components we could build innovative products that solve many daily problems and eliminate the necessity to import ready-made materials from foreign countries, which cost the government heavy losses.

1.2 objectives

First, we will map the vehicle i.e. the car, architecture and set its different properties and merits. This initial stage will help to define deferent necessary electronic components needed for the vehicle assemblage.

Secondly, we will implement the car system responsible for receiving internet-based commands, the operation will entirely on the Arduino hardware. Once the vehicle is assembled and the system is implemented, the testing stage come. This stage is important to ensure that the vehicle works properly and ready to be connected to the remote-operated system.

Next, we will implement the remote-operated system, which is composed of mobile application destined for translating user orders, and server system responsible for Internet-based requests between the mobile application and the vehicle.

Finally, we need to test and examine the final work and express results.

1.3 Outline

The dissertation consists of three chapters with a general introduction and conclusion.

First, the chapter 1 and chapter 2, sequentially, present the literature review related to the internet of things and Arduino hardware with detailed information that helps to understand the dissertation scope and move smoothly to the next chapter.

Next, Chapter 3 presents the practical part of our dissertation. It focuses on the tools used to develop the project with different steps followed until getting final results, which will be examined and tested.

Finally, the conclusion will sum up the dissertation content and evaluate given results, then followed by future work.

CHAPTER 1

INTERNET OF THINGS (IOT)

1 Introduction

The Internet of Things (IoT) is an important topic in technology industry, policy, and engineering circles and has become headline news in both the specialty press and the popular media. This technology is embodied in a wide spectrum of networked products, systems, and sensors, which take advantage of advancements in computing power, electronics miniaturization, and network interconnections to offer new capabilities not previously possible. An abundance of conferences, reports, and news articles discuss and debate the prospective impact of the "IoT revolution" from new market opportunities and business models to concerns about security, privacy, and technical interoperability. IoT systems like networked vehicles, intelligent traffic systems, and sensors embedded in roads and bridges move us closer to the idea of "smart cities'', which help minimize congestion and energy consumption.[1]

2 What is the Internet of Things (IoT)

2.1 Definition

Internet of Things (IoT) is an ecosystem of connected physical objects that are accessible through the internet. The 'thing' in IoT could be a person with a heart monitor or an automobile with built-in-sensors, i.e. objects that have been assigned an IP address and have the ability to collect and transfer data over a network without manual assistance or intervention. The embedded technology in the objects helps them to interact with internal states or the external environment, which in turn affects the decisions taken.[5]

IoT can help organizations reduce cost through improved process efficiency, asset utilization and productivity. With improved tracking of devices/objects using sensors and connectivity, they can benefit from real-time insights and analytics, which would help them make smarter decisions. The growth and convergence of data, processes and things on the internet would make such connections more relevant and important, creating more opportunities for people, businesses and industries.[5]

2.2 A Brief history

The Internet of Things (IoT) has not been around for very long. However, there have been visions of machines communicating with one another since the early 1800s. Machines have been providing direct communications since the telegraph (the first landline) was developed in the 1830 and 1840. Described as "wireless telegraphy," the first radio voice transmission took place on June 3, 1900, providing another necessary component for developing the Internet of Things. The development of computers began in the 1950.[6]

The Internet, itself a significant component of the IoT, started out as part of DARPA (Defense Advanced Research Projects Agency) in 1962, and evolved into ARPANET in 1969. In the 1980s, commercial service providers began supporting public use of ARPANET, allowing it to evolve into our modern Internet. Global Positioning Satellites (GPS) became a reality in early 1993, with the Department of Defense providing a stable, highly functional system of 24 satellites. This was quickly followed by privately owned, commercial satellites being placed in orbit. Satellites and landlines provide basic communications for much of the IoT.[6]

The Internet of Things, as a concept, wasn't officially named until 1999. One of the first examples of an Internet of Things is from the early 1980s, and was a Coca Cola machine, located at the Carnegie Melon University. Local programmers would connect by Internet to the refrigerated appliance, and check to see if there was a drink available, and if it was cold, before making the trip.

By the year 2013, the Internet of Things had evolved into to a system using multiple technologies, ranging from the Internet to wireless communication and from micro-electromechanical systems (MEMS) to embedded systems. The traditional fields of automation (including the automation of buildings and homes), wireless sensor networks, GPS, control systems, and others, all support the IoT. .[6]

2.3 Applications domains

2.3.1 Healthcare

Arguably the greatest technological leap forward in the last several decades has been the growth of electronic health records, or EHRs. In 2009, a mere 16% of U.S. hospitals were using an EHR, but that figure soared to approximately 80% in 2013, according to Becker's Hospital Review.

Previously, hospitals had multiple systems that handled different functions, but EHRs roll all of those into a single system.[7]

Then we have portal technology, which lets patients take a more active role in their own health and well being. Portals let users log on to the healthcare provider's websites to access their medical records, download forms, and prepare for appointments. .[7]

The Internet of Things is slowly starting to weave into healthcare on both the doctor and patient fronts. Ultrasounds, thermometers, glucose monitors, electrocardiograms, and more are all starting to become connected and letting patients track their health. This is crucial for those situations that require follow-up appointments with doctors. .[7]

Multiple hospitals have started to utilize smart beds, which can sense the presence of a patient and automatically adjust itself to the correct angle and pressure to provide proper support without the need for a nurse to intervene. .[7]

The IoT could also help transform patient care at home. Sadly, some patients don't take their medication in appropriate doses or at the correct times. Smart medication dispensers in the home could automatically upload information to the cloud and alert doctors when patients don't take their medicine. More broadly, this type of technology could let doctors know of any potentially dangerous patient behavior. .[7]

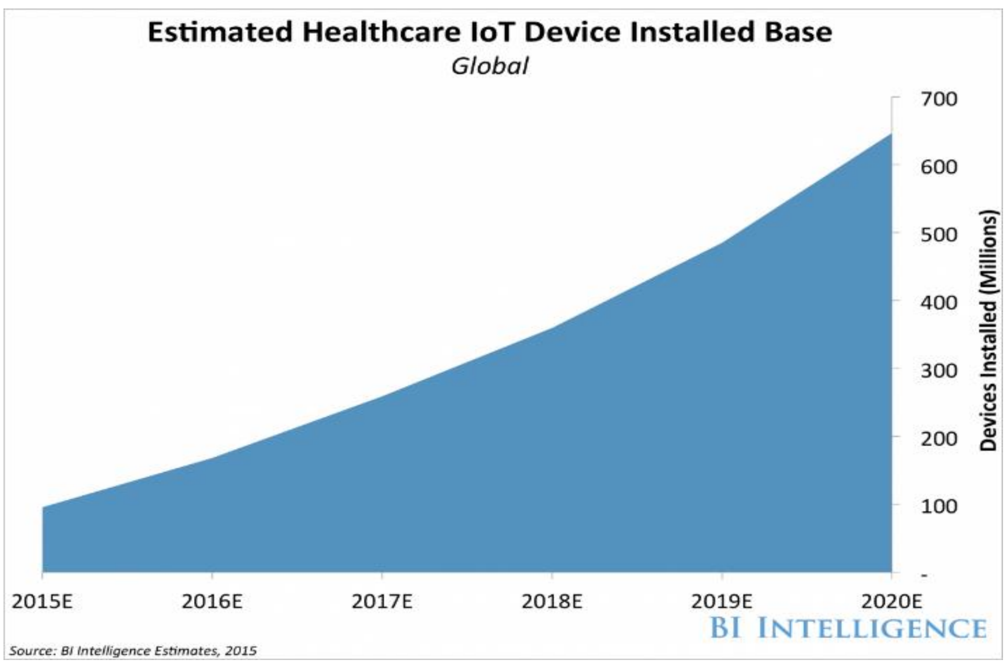

Figure 1.1 Estimated Healthcare IoT Device Installed Base

2.3.2 Information and Communication Technologies (ICT)

The automotive and transportation sector are undergoing tectonic shifts in the capabilities of vehicles and infrastructure as a result of incorporating advanced technologies. Information and communications technology (ICT) such as automotive safety applications, smart fleet management, and intelligent transportation systems (ITS) that are being incorporated into vehicles and infrastructure are key components of the IoT.[8]

The U.S. government should acknowledge and prioritize the ITC sector's value to and impact on the future of America's high-tech, surface transportation ecosystem. ITI is working with U.S. policymakers to align policies that encourage innovation, competition, and private investment, which will be essential for transportation technologies to realize their maximum economic and societal benefits and be broadly available in a timely and globally competitive manner. Research and deployment of advanced ICT-enabled vehicular technologies should be accelerated to save lives, improve quality of life, improve personal and commercial goods mobility, and help address our nation's current and future infrastructure, environmental, and economic challenges.

Smart cities will be the epicenter for a multitude of IoT applications and use cases, including intelligent transportation, smart buildings, improved government services and much more. While these use cases initially will be vertical sector-specific, ultimately our cities will become horizontal IoT platforms where individual use cases can fluidly interconnect to maximize efficiency, productivity and citizen ease-of-use. .[8]

2.3.3 Manufacturing and Heavy Industry (IIOT)

The Industrial Internet of Things encompasses industrial machinery; transportation equipment (cars, trains, and planes); health care equipment; and megasystems like smart buildings, smart cities, and smart utility grids. Their purpose is to increase productivity, allow manufacturers to differentiate through the offering of attached services, and reduce environmental impact. Industrial IoT systems enable applications such as predictive maintenance services, energy optimization, and design optimization.[9]

Manufacturing IoT systems—a subset of industrial IoT systems—include things that are found in a factory for the production of goods (factory buildings, manufacturing equipment, material handling equipment, robots, warehouses, and so forth). Here, manufacturing devices and software are connected to each other to optimize the operations of a factory in real time, synchronize the operation of automated equipment, and optimize the supply chain and inventory management. This is known as "smart manufacturing." Smart manufacturing (or smart factories, as sometimes called) is precisely the goal of Industry 4.0 (the German government initiative). [9]

As the technology matures over time, it will become more apparent and clear as to which IoT lives where and why. In the meantime, there's storming, norming, and seeing what rises to the top of the IoT food chain. But one thing it all comes down to—it's all connected. [9]

2.3.4 Food and Farming

The farming industry will become arguably more important than ever before in the next few decades.

The world will need to produce 70% more food in 2050 than it did in 2006 in order to feed the growing population of the Earth, according to the UN Food and Agriculture Organization. To meet this demand, farmers and agricultural companies are turning to the Internet of Things for analytics and greater production capabilities.[10]

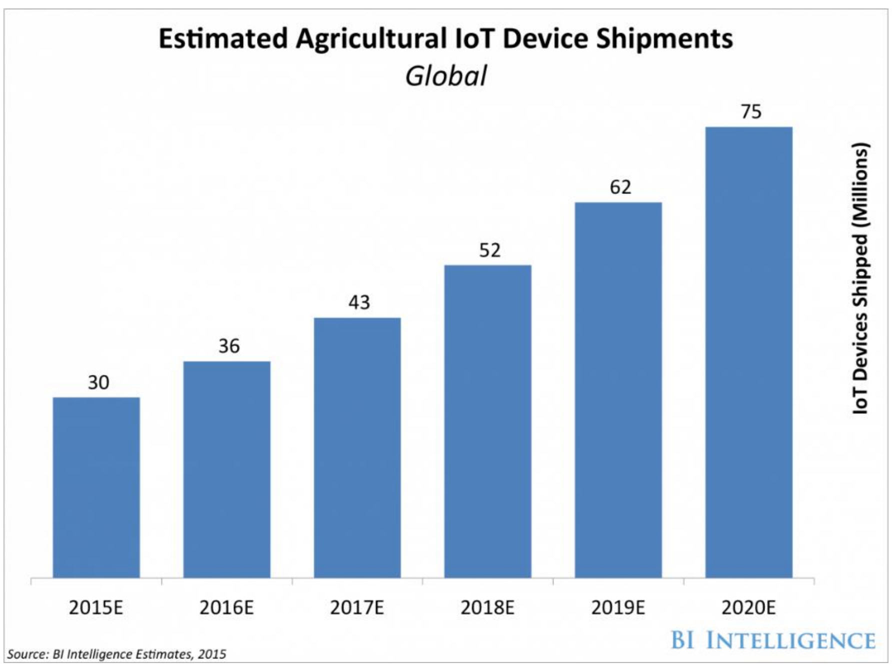

Figure 1.2 Agricultural IoT Device Installed Base

Technological innovation in farming is nothing new. Handheld tools were the standards hundreds of years ago, and then the Industrial Revolution brought about the cotton gin. The 1800s brought about grain elevators, chemical fertilizers, and the first gas-powered tractor. Fast forward to the late 1900s, when farmers start using satellites to plan their work.[10]

The IoT is set to push the future of farming to the next level. Smart agriculture is already becoming more commonplace among farmers, and high tech farming is quickly becoming the standard thanks to agricultural drones and sensors. .[10]

Below, we've outlined IoT applications in agriculture and how "Internet of Things farming" will help farmers meet the world's food demands in the coming years. .[10]

2.3.5 Public Safety and Military

The public safety sector is another area where the IoT could make a real difference, saving lives and even preventing future disasters. [11]

Although it's often discussed as a future technology, the IoT is already here in the numerous devices we use to connect regularly to the Internet and to each other. This will be enhanced in the future as people themselves are able to transmit data as well, due to sensors placed in wearables, sewn into clothing, or even placed directly on the skin at some point. But how does this apply to the realm of public safety?

The constant transfer of data can be used to prime effect in an emergency situation, allowing public safety professionals or first responders to make the lightning-fast decisions that save lives. By using real-time data updates, responses can be better coordinated and processes improved. Here's a closer look at how this operates.

Technology is now a vital component of the public safety industry. Law enforcement officers, firefighters, disaster relief teams, and the military are all entities which use mobile apps, video streaming, and live GPS to coordinate their responses more efficiently.

Public safety agencies are at the same time always under pressure to cut costs and provide more streamlined, efficient operations. Technology is a way to do this, athough it requires some investment. Because the Internet of Things enables machine-to-machine communication, it allows operations to be put into action with minimal personnel. [11]

Emergency response teams need to be able to access important data ideally before they arrive on the scene, while law enforcement must quickly assess a situation that can be both dangerous and fluid. [11]

The first minutes or hours of a disaster situation can be crucial for survivors and first responders alike, but this is when the usual communications infrastructure is most likely to take a hit. The Internet of Things and its attendant technology provides interconnected data that's useful in a wide range of situations.

2.4 Internet of Things Communications Models

At its most basic level, the Internet of Things is all about connecting various devices and sensors to the Internet, but it's not always obvious how to connect them.

In March 2015, the Internet Architecture Board – a group within the Internet Society that oversees the technical evolution of the Internet – released a guide to IoT networking (PDF). This outlined four common communication models used by IoT "smart objects": Device-to-Device, Device-to-Cloud, Device-to-Gateway, and Back-End Data-Sharing.[12]

2.4.1 Device-to-Device Communications

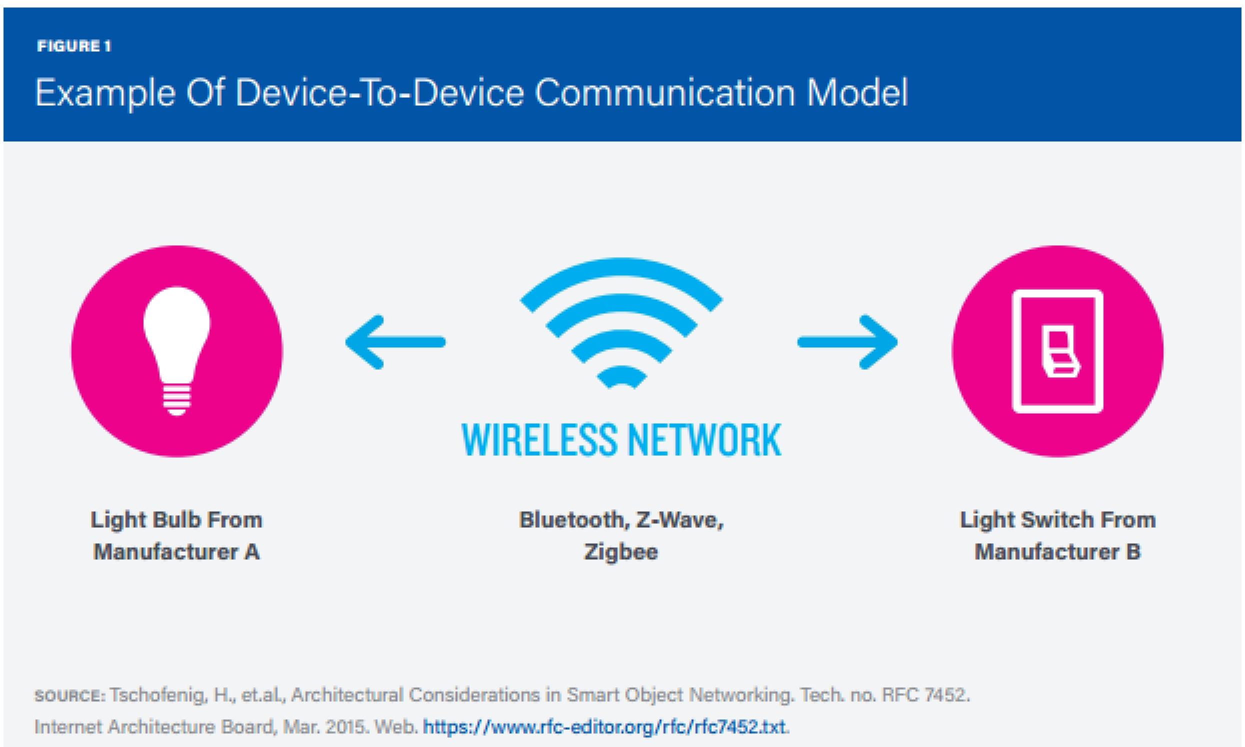

Figure 1.3 Example Device To Device Communication Model

Device-to-device communication represents two or more devices that directly connect and communicate between one another. They can communicate over many types of networks, including IP networks or the Internet, but most often use protocols like Bluetooth, Z-Wave, and ZigBee.

This model is commonly used in home automation systems to transfer small data packets of information between devices at a relatively low data rate. This could be light bulbs, thermostats, and door locks sending small amounts of information to each other. [12]

Each connectivity model has different characteristics, Tschofenig said. With Device-to-Device, he said "security is specifically simplified because you have these short-range radio technology [and a] one-to-one relationship between these two devices." [12]

Device-to-device is popular among wearable IoT devices like a heart monitor paired to a smartwatch where data doesn't necessarily have be to shared with multiple people.

There are several standards being developed around Device-to-Device including Bluetooth Low Energy (also known as Bluetooth Smart or Bluetooth Version 4.0+) which is popular among portable and wearable devices because its low power requirements could mean devices could operate for months or years on one battery. Its lower complexity can also reduce its size and cost.

2.4.2 Device-to-Cloud Communications

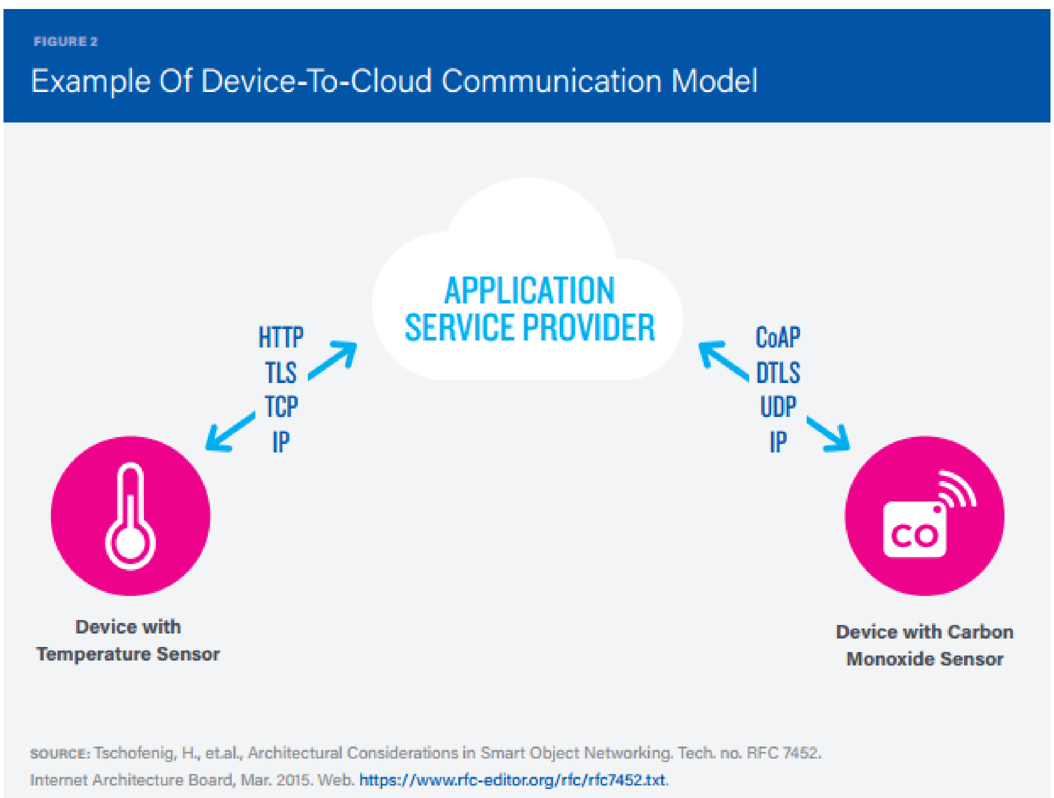

Figure 1.4 Example Device To Cloud Communication Model

Device-to-cloud communication involves an IoT device connecting directly to an Internet cloud service like an application service provider to exchange data and control message traffic. It often uses traditional wired Ethernet or Wi-Fi connections, but can also use cellular technology.

Cloud connectivity lets the user (and an application) to obtain remote access to a device. It also potentially supports pushing software updates to the device. [12]

A use case for cellular-based Device-to-Cloud would be a smart tag that tracks your dog while you're not around, which would need wide-area cellular communication because you wouldn't know where the dog might be. [12]

Another scenario, Tschofenig said, would be remote monitoring with a product like the Dropcam, where you need the bandwidth provided by Wifi or Ethernet. But it also makes sense to push data into the cloud in this scenario because makes sense because it provides access to the user if they're away. "Specifically, if you're away and you want to see what's on your webcam at home. You contact the cloud infrastructure and then the cloud infrastructure relays to your IoT device."

From a security perspective, this gets more complicated than Device-to-Device because it involves two different types of credentials – the network access credentials (such as the mobile device's SIM card) and then the credentials for cloud access. [12]

The IAB's report also mentioned that interoperability is also a factor with Device-to-Cloud when attempting to integrate devices made by different manufacturers given that the device and cloud service are typically from the same vendor. An example would be the Nest Labs Learning Thermostat, where the Learning Thermostat can only work with Nest's cloud service.

Tschofenig said there's work going into making Wifi devices that make cloud connections while consuming less power with standards such as LoRa, Sigfox, and Narrowband. [12]

2.4.3 Device-to-Gateway Model

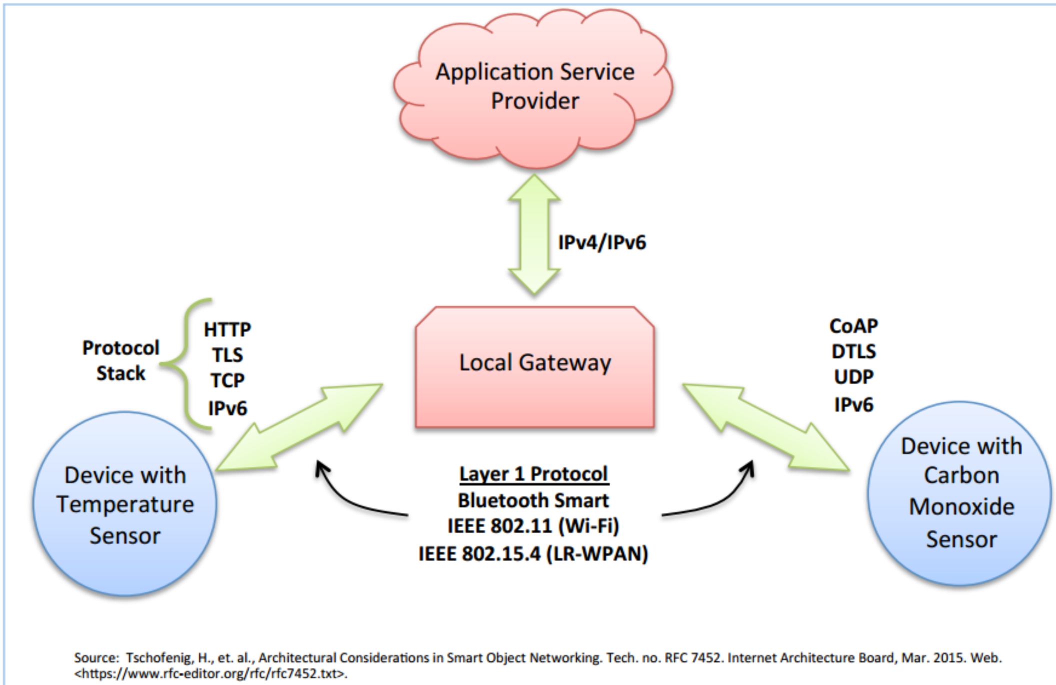

Figure 1.5 Example Device To Gateway Communication Model

In the Device-to-Gateway model, IoT devices basically connect to an intermediary device to access a cloud service. This model often involves application software operating on a local gateway device (like a smartphone or a "hub") that acts as an intermediary between an IoT device and a cloud service.

This gateway could provide security and other functionality such as data or protocol translation. If the application-layer gateway is a smartphone, this application software might take the form of an app that pairs with the IoT device and communicates with a cloud service.

This might be a fitness device that connects to the cloud through a smartphone app like Nike+, or home automation applications that involve devices that connect to a hub like Samsung's SmartThings ecosystem.

"Today, you more or less have to more or less buy a gateway from a dedicated vendor or use one of these mulit-purpose gateways," Tschofenig said. "You connect all your devices up to that gateway and it does something like data aggregation or transcoding, and it either hands [off the data] locally to the home or shuffles it off to the cloud, depending on the use case."

Gateway devices can also potentially bridge the interoperability gap between devices that communicate on different standards. For instance, SmartThings' Z-Wave and Zigbee transceivers can communicate with both families of devices. [12]

2.4.4 Back-End Data-Sharing Model

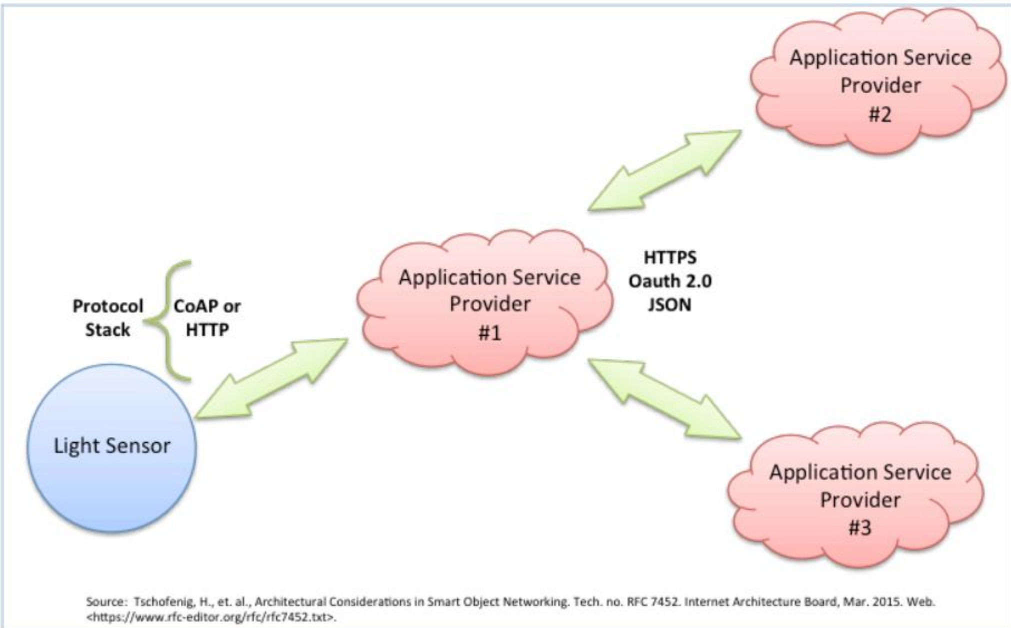

Figure 1.6 Example Back-End Data-Sharing Communication Model

The back-end data-sharing model refers to a communication architecture that enables users to export and analyze smart object data from a cloud service in combination with data from other sources.

This architecture supports "the [user's] desire for granting access to the uploaded sensor data to third parties".50 This approach is an extension of the single device-to-cloud communication model, which can lead to data silos where "IoT devices upload data only to a single application service provider''.51 A back-end sharing architecture allows the data collected from single IoT device data streams to be aggregated and analyzed. [12]

For example, a corporate user in charge of an office complex would be interested in consolidating and analyzing the energy consumption and utilities data produced by all the IoT sensors and Internet-enabled utility systems on the premises. Often in the single device-to-cloud model, the data each IoT sensor or system produces sits in a stand-alone data silo.

An effective back-end data sharing architecture would allow the company to easily access and analyze the data in the cloud produced by the whole spectrum of devices in the building. Also, this kind of architecture facilitates data portability needs. Effective back-end datasharing architectures allow users to move their data when they switch between IoT services, breaking down traditional data silo barriers. The back-end data-sharing model suggests a federated cloud services approach52 or cloud applications programmer interfaces (APIs) are needed to achieve interoperability of smart device data hosted in the cloud. [12]

This architecture model is an approach to achieve interoperability among these back-end systems. As the IETF Journal suggests, "Standard protocols can help but are not sufficient to eliminate data silos because common information models are needed between the vendors."54 In other words, this communication model is only as effective as the underlying IoT system designs. Back-end data sharing architectures cannot fully overcome closed system designs.

2.5 Internet of Things Communications Models Summary

The four basic communication models demonstrate the underlying design strategies used to allow IoT devices to communicate. Aside from some technical considerations, the use of these models is largelyinfluenced by the open versus proprietary nature of the IoT devices being networked. And in the case of the device-to-gateway model, its primary feature is its ability to overcome proprietary device restrictions in connecting IoT devices. [12]

This means that device interoperability and open standards are key considerations in the design and development of internetworked IoT systems.From a general user perspective, these communication models help illustrate the ability of networked devices to add value to the end user. By enabling the user to achieve better access to an IoT device and its data, the overall value of the device is amplified.

For example, in three of the four communication models, the devices ultimately connect to data analytic services in a cloud computing setting. By creating data communication conduits to the cloud, users, and service providers can more readily employ data aggregation, big data analytics, data visualization, and predictive analytics technologies to get more value out of IoT data than can be achieved in traditional data-silo applications.

In other words, effective communication architectures are an important driver of value to the end user by opening possibilities of using information in new ways. It should be noted, however, these networked benefits come with trade offs.

Careful consideration needs to be paid to the incurred cost burdens placed on users to connect to cloud resources when considering an architecture, especially in regions where user connectivity costs are high.While the end user benefits from effective communication models, it should be mentioned that effective IoT communication models also enhance technical innovation and open opportunity for commercial growth. New products and services can be designed to take advantage of IoT data streams that didn't exist previously,acting as a catalyst for further innovation. [12]

2.6 The Impact of IoT on Society

The Internet has (traditionally) been seen to be about connecting people and information. We are now dealing with a level of abstraction beyond that. However, unlike the Internet, which is a concrete technical infrastructure whose design and architecture are well documented, the IoT is still primarily a vision and only part reality individual IoT technologies and systems exist, but there is currently no coherent global IoT. Whatever form it takes, the IoT will most likely be an extension of the Internet; distinctions between the IoT and the Internet are therefore often difficult to maintain, and they raise similar issues and challenges, for example, surrounding privacy and data protection.[13]

That said, the IoT introduces newchallenges, carrying with it an inherent assumption that information will be shared across things, applications and possibly sectors. This data-sharing assumption might lead to the IoT having even more dramatic impacts on privacy and data protection than other Information and Communication Technologies (ICTs); such as when energy or water meter readings are used to alert a family about the health of an elderly relative living alone, or when people are tracked, making them part of the IoT.

The IoT also brings with it a new scale of development. There are fewer than 10 billion people on the planet, but there could be a trillion sensor devices. Worryingly, there appears to be a lack of appreciation of this scale, and the pervasiveness of its potential application across all sectors of society. While current IoT applications are still very traditional, there is likely be more radical, emergent, unpredictable, and user-led innovation in the future; just as we have seen with the Internet. [13]

2.7 Advantage and disadvantages of IoT

2.7.1 Advantages

Automation of daily tasks leads to better monitoring of devices

The IoT allows you to automate and control the tasks that are done on a daily basis, avoiding human intervention. Machine-to-machine communication helps to maintain transparency in the processes. It also leads to uniformity in the tasks. It can also maintain the quality of service. We can also take necessary action in case of emergencies.

Efficient and Saves Time

The machine-to-machine interaction provides better efficiency, hence; accurate results can be obtained fast. This results in saving valuable time. Instead of repeating the same tasks every day,it enables people to do other creative jobs.

Saves Money

Optimum utilization of energy and resources can be achieved by adopting this technology and keeping the devices under surveillance. We can be alerted in case of possible bottlenecks, breakdowns, and damages to the system. Hence, we can save money by using this technology.

All the applications of this technology culminate in increased comfort, convenience, and better management, thereby improving the quality of life.

2.7.2 Disadvantages

Loss of privacy and security

As all the household appliances, industrial machinery, public sector services like water supply and transport, and many other devices all are connected to the Internet, a lot of information is available on it. This information is prone to attack by hackers. It would be very disastrous if unauthorized intruders access private and confidential information.

Compatibility

As devices from different manufacturers will be interconnected, the issue of compatibility in tagging and monitoring crops up. Although this disadvantage may drop off if all the manufacturers agree to a common standard, even after that, technical issues will persist. Today, we have Bluetooth-enabled devices and compatibility problems exist even in this technology! Compatibility issues may result in people buying appliances from a certain manufacturer, leading to its monopoly in the market.

Complexity

The IoT is a diverse and complex network. Any failure or bugs in the software or hardware will have serious consequences. Even power failure can cause a lot of inconvenience.

Lesser Employment of Menial Staff

The unskilled workers and helpers may end up losing their jobs in the effect of automation of daily activities. This can lead to unemployment issues in the society. This is a problem with the advent of any technology and can be overcome with education.

Technology Takes Control of Life

Our lives will be increasingly controlled by technology, and will be dependent on it. The younger generation is already addicted to technology for every little thing. We have to decide how much of our daily lives are we willing to mechanize and be controlled by technology.

3 Latest news about IoT

3.1 Drones And The Internet Of Things

All drones are not equal. Some like the Global Hawk are very complex systems that are connected to satellites and are only the purview of the military.

Others like the Parrot A.R. Drone are mass-produced hobby aircraft that you can control with your mobile device. But a class of drones in the middle combines the capabilities of both complex and mass-produced systems and is specifically designed for commercial purposes. These drones weight less than 55 lbs. and are classified by regulatory entities as small-unmanned aircraft systems or sUAS. We don't see their ubiquitous use in the U.S. quite yet, but in countries like England, Australia, and France, you will find them operating in energy, mining, mapping, and surveying companies – and quite a few government agencies like those responsible for transportation and infrastructure.[14]

Commercial drones are truly 'unmanned aircraft systems'. They are not just remote controlled aircraft. They require many things in order to run, like avionics, ground control stations, communication systems, data collection and processing software, and of course GPS for geo-referencing. There's more, but you get the idea. These are multifaceted complex vehicles whose mission is to fly sensors and collect data.

Commercial drones are also connected devices. So they are ‘things in motion’. Most are accessible or controllable over the Internet, and the data they collect is pushed to various cloud services. Some drones are beginning to carry on-board processors as well and are now part of the growing trend of fog computing devices.[14]

3.2 Wearable Devices

CCS Insight expects the wearables market to reach $14 billion by the end of this year, according to Forbes. And BI Intelligence, Business Insider's premium research service, expects the wearables market to grow to 162.9 million units by the end of 2020.

The healthcare sector will be one of the top catalysts to push the wearables markets to these heights, as consumer and professional healthcare trends will spur interest in wearable devices.

Fitness trackers, in particular, are the leading consumer case for wearables because most consumers use wearable devices to record their exercise and health statistics and progress. And hospitals, med-tech companies, pharmaceutical companies, and insurance companies have started to utilize these devices.[15]

Accuracy will remain the top barrier to widespread adoption, as manufacturers must ensure that these devices transmit correct data so that users receive accurate progress reports. And as always, privacy concerns are of the utmost importance.[15]

Furthermore, smartwatches will likely cannibalize a major portion of demand for fitness trackers, as these devices perform many of the same functions as fitness bands but carry a host of other features, as well.

But other questions still remain. Will these devices truly be transformative? Will the average consumer want to have a wearable device attached to them 24 hours a day to monitor all of their activities? Will these devices need to be more fashion friendly, or even totally concealed? Will the government impose restrictive regulations on IoT devices?[15]

4 Predictions for the Future of IoT

Smart tech will quickly become the backbone of retail, although it may be in ways the customer doesn’t always see. While there will invariably be beautiful, innovative experiences that involve products and fixtures with instrumentation that make the store “come alive” around the shopper in a highly visible and explicit way, the real revolution on the horizon is around “implicit” smart technology. The instrumentation in the stores that constantly assess products, people, and productivity and directs staff on how to optimize the store so the customer finds less and less friction in their shopping journey. Things like high-fidelity IR arrays and UHF RFID will quickly prove data in physical stores is far better than online… once smart tech is at the helm.[16]

As IoT gains momentum, the volume of data generated will be stratospheric. Not only will there be more data, but there will be different types of data, and data from sources that have yet to be considered. Big Data analytics will evolve into a distributed analytics model, which will help with the monetization of IoT data. We will see more devices capable of analyzing data locally, processing and capturing the most important data for more real-time IoT services.[16]

More and more IoT devices, as well as a wider variety of devices, will enter the market that are IP enabled. In the short term this is going to create additional vulnerabilities and present challenges to security professionals across vastly enlarged attack surface everywhere from homes to enterprises to even automobiles.[16]

With higher value and higher consequence devices, like those found in an automobile, automakers will start to pay closer attention to security in a number of ways, from encrypting and securing control planes like CANbus–which previously were assumed to be secure via obscurity–as well as wider use of OTA updates. Lower value devices, including IP cameras, routers, among others, will eventually have more widely available updates, but these will be mostly manual, pull updates. It is likely we will see more automated push updates may be made available later to help better secure a wide spectrum of consumer devices, at least to combat well known and documented threats.[16]

CHAPTER 2

MICROCONTROLLERS AND ARDUINO BOARD

1 Introduction

The Arduino environment has been designed to be easy to use for beginners who have no software or electronics experience. With Arduino, you can build objects that can respond to and/or control light, sound, touch, and movement. Arduino has been used to create an amazing variety of things, including musical instruments, robots, light sculptures, games, interactive furniture, and even interactive clothing.

Though it is easy to use, Arduino's underlying hardware works at the same level of sophistication that engineers employ to build embedded devices. People already working with microcontrollers are also attracted to Arduino because of its agile development capabilities and its facility for quick implementation of ideas.

2 MICROCONTROLLERS

2.1 What is the microcontroller?

A microcontroller is a compact microcomputer designed to govern the operation of embedded systems in motor vehicles, robots, office machines, complex medical devices, mobile radio transceivers, vending machines, home appliances, and various other devices. A typical microcontroller includes a processor, memory, and peripherals.

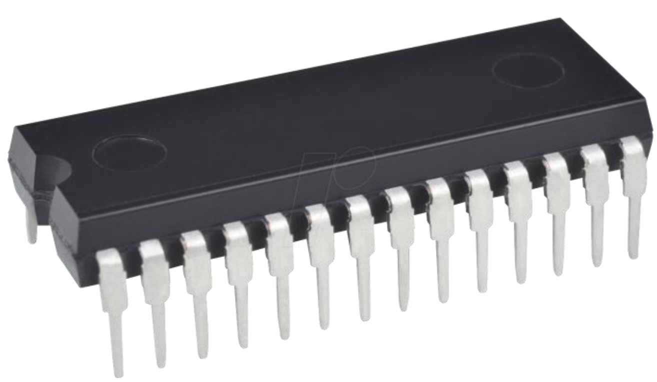

Figure 2.1 Microcontroller

The simplest microcontrollers facilitate the operation of the electromechanical systems found in everyday convenience items. Originally, such use was confined to large machines such as furnaces and automobile engines to optimize efficiency and performance.

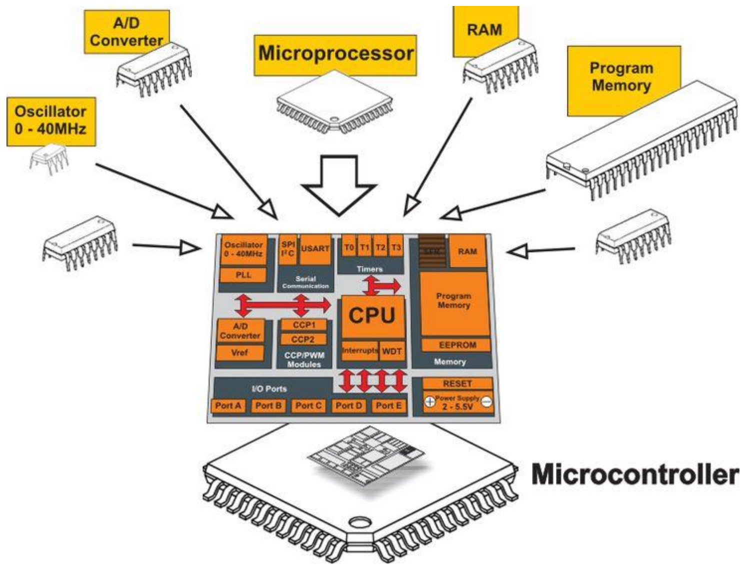

Figure 2.2 Microcontroller architecture

In recent years, microcontrollers have found their way into common items such as ovens, refrigerators, toasters, clock radios, and lawn watering systems. Microcomputers are also common in office machines such as photocopiers, scanners, fax machines, and printers.

2.2 Uses of microcontroller

2.2.1 Microcontrollers in Health Care

Products that involve an electronic-human interface make use of a microcontroller. A microcontroller has the capability to interface with a keyboard, accept information and display outputs on a screen for viewing. Microcontrollers can be used in basic instruments from calculators to high-end sophisticated health care products such as heart monitors. Almost all electronic medical devices such as blood pressure monitors, blood sugar meters and blood oxygen saturation meters have advanced microcontrollers.

2.2.2 Microcontrollers in the Defense Sector

Sophisticated electronic weapons used in the defense sector also contain microcontrollers. Anti-tank missiles, surface-to-air missiles and now even guns used by the military have microcontrollers in their electronic circuitry. Microcontrollers are used as decision-making and computational devices that simplify the use of instruments for humans.

2.2.3 Microcontrollers in Toys

Electronic interactive toys found in everyday life contain microcontrollers. Toys have been made more fun and have come to provide a huge educational platform for kids due to the integration of microcontrollers. Mini-robot toys, remote controlled cars, helicopters and planes are some of the examples of products that make use of microcontrollers.

2.3 Difference between microprocessor and microcontroller

Microprocessor is an IC which has only the CPU inside them i.e. only the processing powers such as Intel's Pentium 1,2,3,4, core 2 duo, i3, i5 etc. These microprocessors don't have RAM, ROM, and other peripheral on the chip. A system designer has to add them externally to make them functional. Application of microprocessor includes Desktop PC's, Laptops, notepads etc.

But this is not the case with Microcontrollers. Microcontroller has a CPU, in addition with a fixed amount of RAM, ROM and other peripherals all embedded on a single chip. At times it is also termed as a mini computer or a computer on a single chip. Today different manufacturers produce microcontrollers with a wide range of features available in different versions. Some manufacturers are ATMEL, Microchip, TI, Freescale, Philips, Motorola etc .

Microcontrollers are designed to perform specific tasks. Specific means applications where the relationship of input and output is defined. Depending on the input, some processing needs to be done and output is delivered. For example, keyboards, mouse, washing machine, digicam, pendrive, remote, microwave, cars, bikes, telephone, mobiles, watches, etc. Since the applications are very specific, they need small resources like RAM, ROM, I/O ports etc and hence can be embedded on a single chip. This in turn reduces the size and the cost.

3 Arduino board

3.1 What is Arduino?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.[17]

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike .[17]

3.2 A Brief history

In 2005, in Ivrea, Italy, a project was initiated to make a device for controlling student-built interactive design projects that was less expensive than other prototyping systems available at the time. One of the cofounders, Massimo Banzi, named this piece of hardware Arduino in honor of Bar di Re Arduino (In 1002, King Arduin became the ruler of the Italy. Today, the Bar di Re Arduino, a pub on a cobblestoned street in town, honors his memory), and began producing boards in a small factory located in the same region as the computer company Olivetti.[17][18]

The Arduino project is a fork of the open source Wiring platform and is programmed using a Wiring-based language (syntax and libraries), similar to C++ with some slight simplifications and modifications, and a Processing-based integrated development environment (IDE).[18]

Arduino was built around the Wiring project of Hernando Barragan. Wiring was Hernando's thesis project at the Interaction Design Institute Ivrea. It was intended to be an electronic version of Processing that used our programming environment and was patterned after the Processing syntax. Arduino would not exist without Wiring and Wiring would not exist without Processing.

3.3 What can I do with an Arduino?

Arduino has huge flexibility with which you can make almost anything you imagine. It can be easily connected to a variety of modules like fire sensors, obstacle sensors, presence detectors, GPS modules, GSM Modules or anything with which you wish to give wings to your dream project. You can find many Arduino projects in the Arduino hacks section in DIY Hacking.

3.4 Arduino architecture

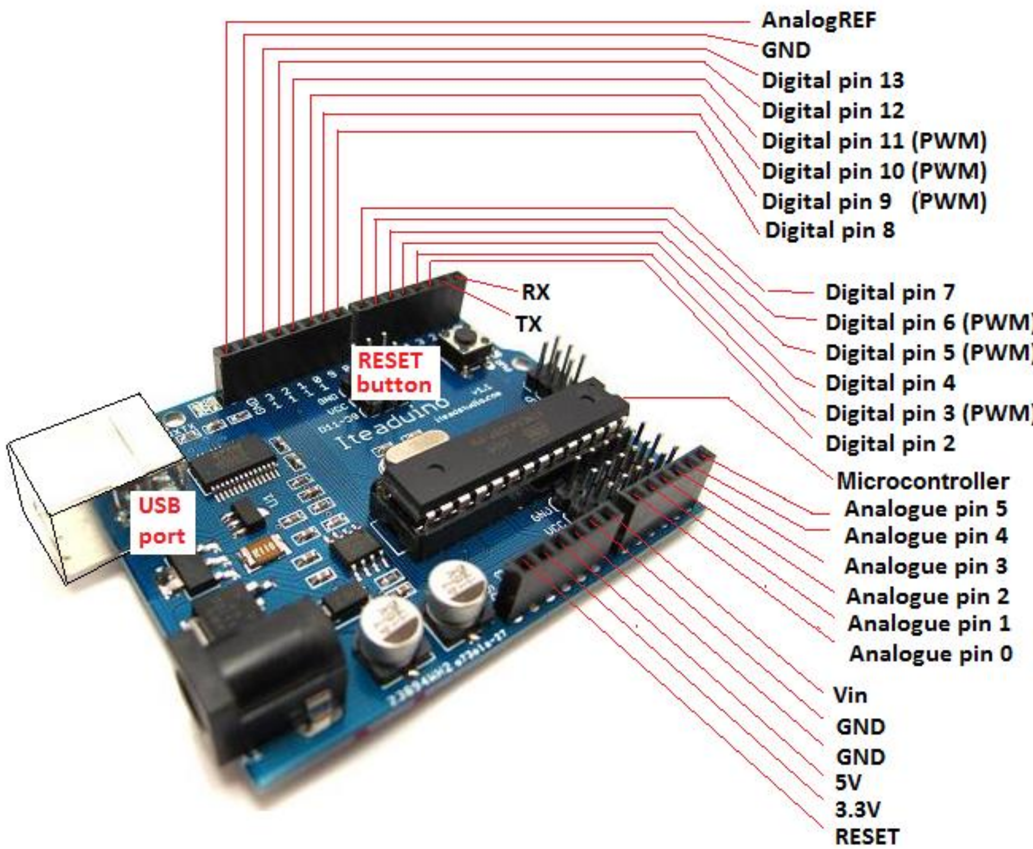

Figure 2.3 Microcontroller architecture

3.4.2 Microcontroller

Each Arduino board has its own microcontroller ,You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC.[19]

Arduino Uno use AT328p Microcontroller, The ATmega328 has 32 KB (with 0.5 KB occupied by the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).[19][2]

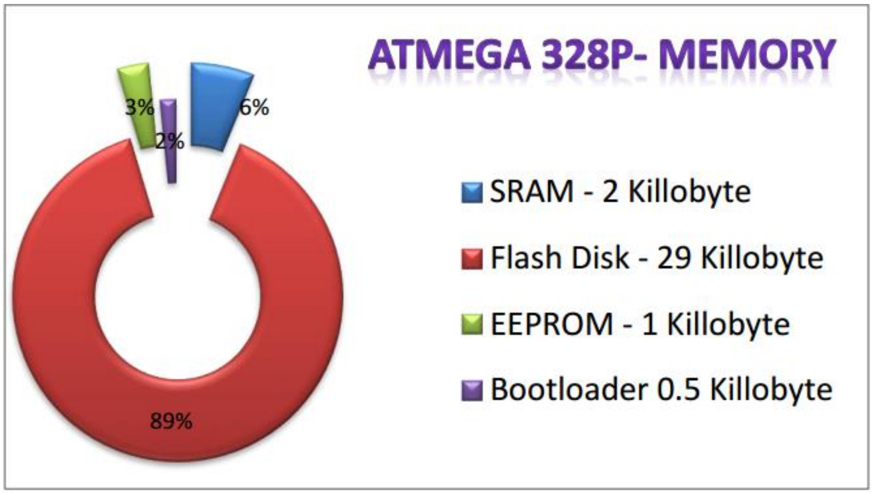

Figure 2.4 ATMEGA 328P architecture

SRAM : Memory used to store variables temporary

FLASH DISK : A storage space used to store the program

EEPROM : Memory used to store some variables permanently

BOOTLOADER : The software responsible how the circuit understands Arduino C language

3.4.3 Arduino Pins

3.4.3.1 Power Pins

The power pins are as follows:

- **Vin :** The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V:** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

- **3V3:** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND:** Ground pins.
- **IOREF:** This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.

3.4.3.2 Digital I/O Pins

The Arduino UNO board has 14 digital I/O pins (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc.

3.4.3.3 Analog Pins

The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

3.4.3.4 Pulse width modulation (PWD)

Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width. To get varying analog values, you change, or modulate, that pulse width. If you repeat this on-off pattern fast enough with an LED for example, the result is as if the signal is a steady voltage between 0 and 5v controlling the brightness of the LED.

3.5 Arduino Board Types

These boards are the backbone to an Arduino build and contain the processors, memory, input, and output ports. There are many from which to choose. The Uno and Leonard are underpowered for bigger, more ambitious builds, but they make a good starting place for those wanting to learn.

Features	Arduino Uno	Arduino Due	Arduino Mega	Arduino Leonardo
Processor	16Mhz ATmega328	84MHz AT91SAM3X8E	16MHz ATmega2560	16MHz ATmega32u4
Memory	2KB SRAM, 32KB flash	96KB SRAM, 512KB flash	8KB SRAM, 256KB flash	2.5KB SRAM, 32KB flash
Digital I/O	14	54	54	20
Analogue I/O	6 input, 0 output	12 input, 2 output	16 input, 0 output	12 input, 0 output

Figure 2.5 Arduino boards features

Different boards support the use of different Arduino shields. These shields are add-ons that add features to the board. Many accessories allow extra USB connectors or screens, which most of the boards support.

3.5.2 The Arduino Uno

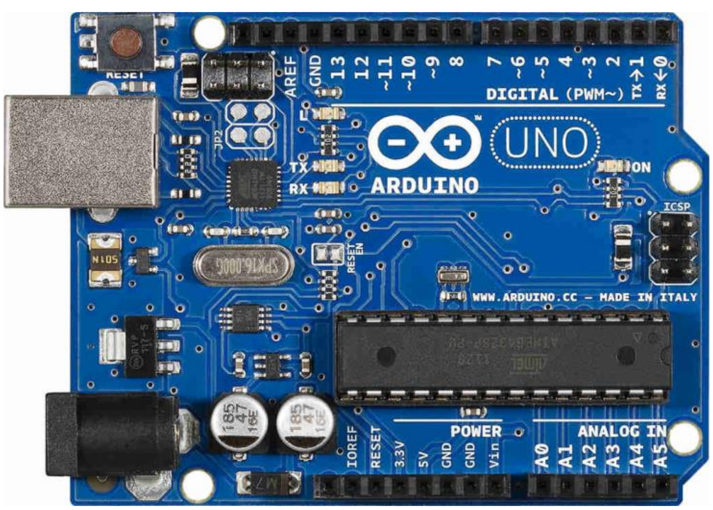

Figure 2.6 Arduino Uno

The Uno is the most common board and the one labeled as the classic Arduino. This board comes with everything new users need to learn about the electronics and programming required to start this hobby. It is compatible with most available Arduino shields.

3.5.3 The Arduino Due

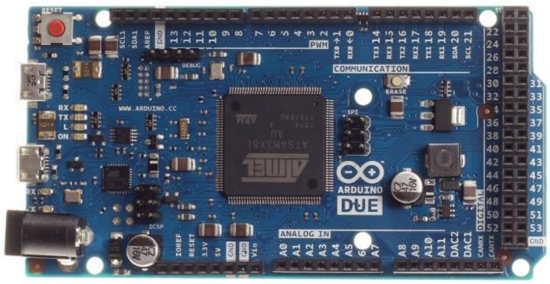

Figure 2.7 Arduino Due

The Arduino Due is the second iteration of the classic Arduino and offers more features for advanced users. The Due's processor is faster, has more memory, and more I/O ports. It does not support many shields. Because of the faster CPU, the Arduino Due runs on a lower voltage: 3.3V over the Uno's 5V. This means it cannot always support the same devices.[20]

3.5.4 The Arduino Mega

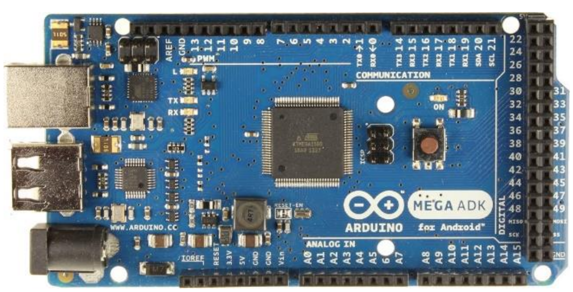

Figure 2.8 Arduino Mega

The Arduino Mega comes in two types, the Mega 2560 and the MEGA ADK. The ADK is similar to the 2560; however, it also has a programmable USB host chip installed. It uses the same 5V power supply as the Uno, so many of the Arduino shields are also compatible with the Mega; however, because of the placement of some of the pins, not all of them are usable. [20]

3.6 The Arduino Shields

Arduino Shields are boards that connect to a number of different Arduino models. They extend the abilities of the basic board by adding features such as wireless network access, cell access, or the ability to prototype circuits. [20]

3.6.1 Wireless Shields

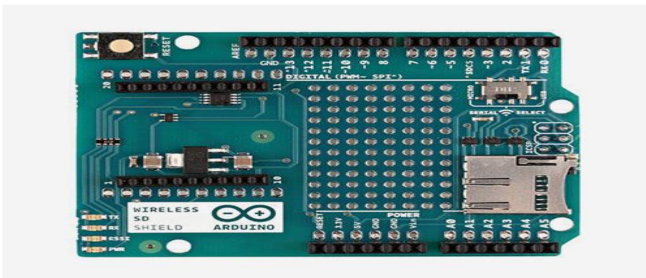

Figure 2.9 Wireless Shield

The wireless-enabled shields for the Arduino come in two types. One, the Wi-Fi Shield, allows the board to access the Internet through an 802.11 b/g-supported network and has a built-in micro-SD card slot to host files accessible through the Internet or the network. The Wireless SD Shield has an XBee module and enables communications between the Arduino and other XBee-supported devices, including the Wireless Proto Shield. [20]

3.6.2 The GSM Shield

Figure 2.10 GSM Shield

The GSM Shield has a slot for a cell phone SIM card. This allows users to create an alert system where they would receive a text or phone call from the Arduino. The shield could also allow performing functions when it received a call or a text from the user. [20]

3.6.3 The Ethernet Shield

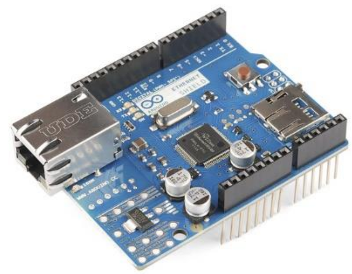

Figure 2.11 Ethernet Shield

The Ethernet Shield is much like the Wi-Fi Shield but supports connections to the network through Ethernet or network cables instead. It also features an SD slot to host files on the network or the Internet. [20]

3.7 Connecting Arduino to PC

Before you start your Arduino projects, you need to setup the connection between your PC and Arduino board. The setup is simple all you need is :

- Arduino board

  You can use any type of Arduino boards

- USB cable (A plug to B plug)
- Arduino Environment Software (IDE)

  You can use any type of Arduino boards

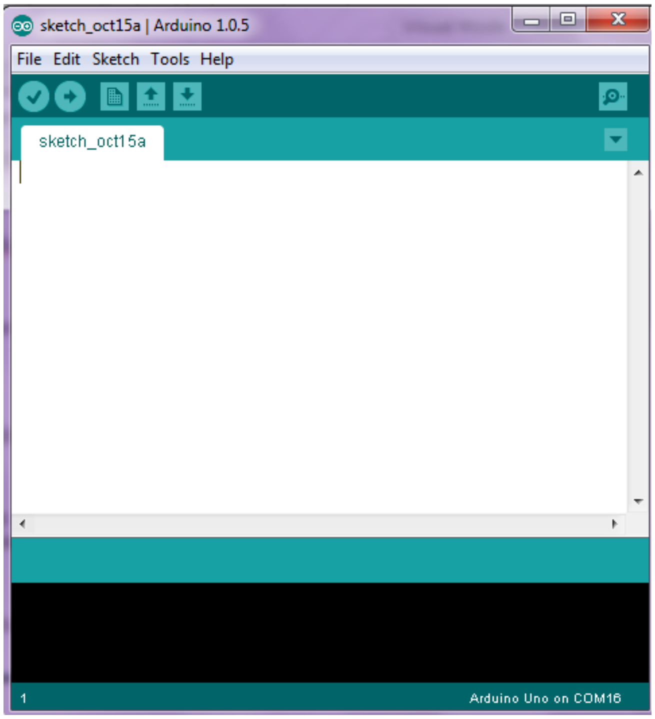

Figure 2.12 Arduino IDE

3.8 Arduino on the software side

The Arduino software is open-source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL. The Arduino program called "sketch".

3.8.1 Arduino IDE (Development Environment)

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board.

3.8.2 Arduino Program Structure

Arduino programs can be divided in three main parts: Structure, Values (variables and constants), and Functions.

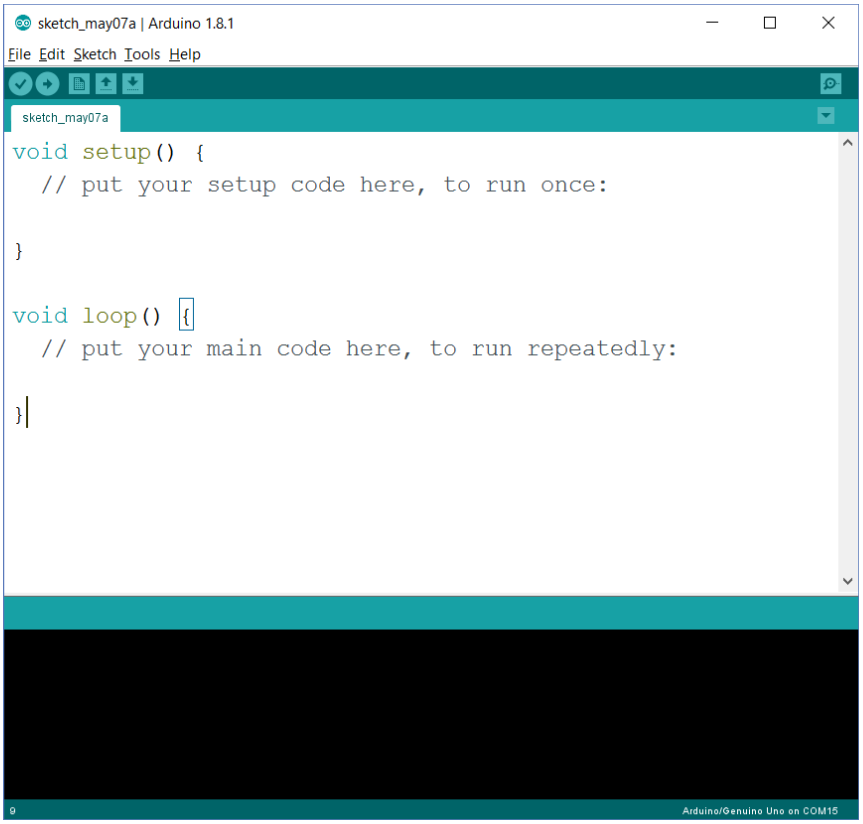

Figure 2.13 Arduino Program Structure

The Structure. Software structure consist of two main functions **Setup( )** function **Loop( )** function.

The **setup()** function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.

After creating a **setup()** function, which initializes and sets the initial values, the loop() function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.

IMPLEMENTATION

This chapter presents the practical part from our dissertation. We will deeply deal with the different stages followed until the completion of the project. We can divide the developed system into three parts, namely: 1st is the ground vehicle, 2nd the server system, 3rd the remote control presented as Android-based mobile application.

For the ground vehicle, it is a mobile robot manipulated remotely based of requests received via Internet. The received requests are presented in four orders as well as connection and disconnection order, the four orders are: order to move forward, order to move back, order to move right and order to move left.

The server system is responsible on processing the connect/disconnect requests come from the mobile device, as well as, moving requests transferring from mobile device to the vehicle, and transferring positions information from the vehicle to the mobile device. All these operations will be in real time.

The mobile application will be like the remote control, it enables for the user to control the vehicle remotely and, simultaneously, receive positioning information from the vehicle in real time.

To make understanding easier to the reader, we expressed the building and implementation of the project exactly same to the real steps that we followed in experimental. We will define the different physical components, especially, the vehicle architecture. Then, we will explain the techniques followed to implement the soft side, where we will talk about algorithms used for each project's component.

In the next section we will test, experimentally, the entire system, including all provided operations.

1 System architecture

In this section, our explanation affects the physical architecture of the vehicle, as well as the implemented systems of each physical part.

1.1 Vehicle architecture

The vehicle body contains the platform which we will install the Arduino with its modules and breadboard for wiring between deferent robot component.

It also contains two big fixed wheels on the left and right side connected to two separate motors that are responsible for the ground movements, as well as, there is a small axial mobile wheel i.e. able for fully rotation. Moving two sides motors in the opposite direction with the small axial wheel are responsible for the movement toward the right and left directions.

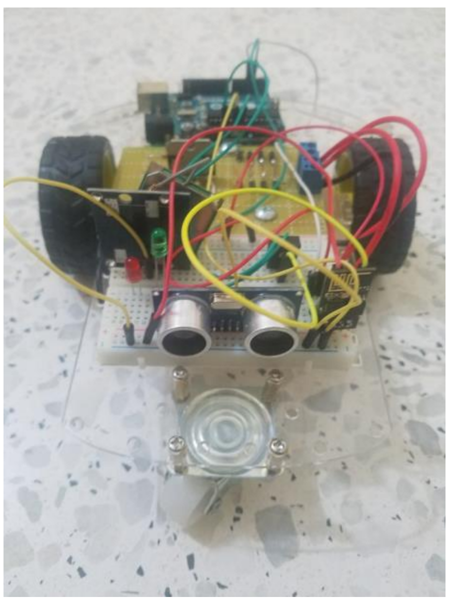

Figure 2.14 Vehicle

1.1.1 Arduino circuit board

The Dev boards are the principle responsible for the robots control, they can be considered as robot brain, they perform all the operations related to robots like sending commands to the other components and receive information from them. Every Dev board has a programming interface that specify the robot behaviors

At the first stage from this project, we found many types of circuit boards with different characteristics and different prices, the table below shows a comparison between a bunch of circuit boards including the chosen circuit board for our project i.e. Arduino:

			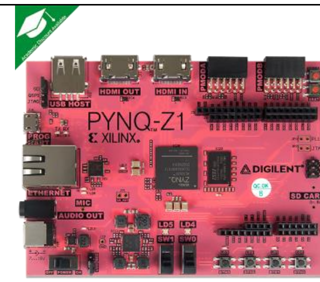
Name	Arduino	Raspberry pi 3 model b	FPGA
Manufacture company	Open Source	Raspberry pi	Xilinx
Features	- Microcontroller: ATmega328P - Operating Voltage:5V - Input Voltage :7-12V - Input Voltage:6-20V - Digital I/O Pins - PWM Digital - DC Current per I/O Pin - DC Current for 3.3V Pin - Flash Memory :32 KB (ATmega328P) - SRAM :2 KB (ATmega328P) - EEPROM :1 KB (ATmega328P) - Clock Speed :16 MHz	- 1GB RAM - 4 USB ports - Full HDMI port - Ethernet port - Camera interface - Display interface - Micro SD card slot	- 650MHz dual-core Cortex-A9 processor 512MB DDR3 with 16-bit bus @ 1050Mbps - MicroSD slot - Powered from USB or any 7V-15V source - Ethernet port - HDMI source port
Price starts from	22$	35$	100$

Figure 2.15 comparison between dev boards

According to the previews table, the Arduino board has less characteristics comparing to Raspberry and FPGA. However, the price of the Arduino board is the most cheaper than both, as well as, it covers all the essential requirements for our project that is:

It is open source

It can communicate with computers via serial connection USB

It can be powered from USB or DC

It can process information without the need of external powerful system such as computers, powered by chip programmable and it has its own memory.

It can work with both digital & analog electronics signal, sensor and actuators

Libraries can be written in C/C++

According to what we have seen in the literature chapter, there are many Arduino types based on the type of the microcontroller speed used and the number of digital and analog inputs and outputs pins.

We selected the Arduino Uno i.e. has an ATmega328 microcontroller board and 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, 16 MHz crystal oscillator, USB connection, power jack, ICSP header, and reset button, because its microcontroller speed is sufficient for our project, and the I/O are sufficient to connect all robot components and modules that we need, moreover, it has a strong resistance to high tensions.

1.1.2 modules

Modules are components used to connect to the Arduino and interact with, they are differentiating according to their roles and functions. Their main role is to add new features to the Arduino. In our project, we used three modules Wi-Fi module, GPS module and ultrasonic module.

1.1.2.1 Wi-Fi module

Figure 2.16 Wifi module esp8266 version 01

At first attempt, we have faced undetermined problems when we used the version 07 of the ESP8266 Wi-Fi module, the module was not able to connect to the Arduino and we failed to identify the problem, so we used version 01, which worked perfectly and gave successful results. This module can work as a access point or client, it plugged to the serial communication (Tx/Rx).

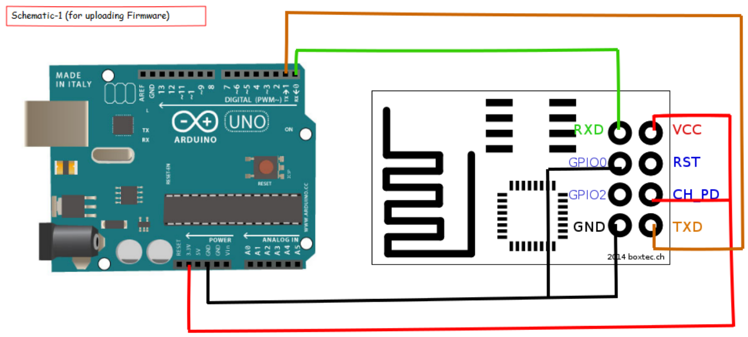

Figure 2.17 Wifi module wiring

When the Wi-Fi module is successfully connected, a red led light will be ignited.

We configured the module via sending commands to the serial port connected to it. The ESP8266 Wi-Fi module, by default, boots up into the serial modem mode, this mode will allow to perform communication using a set of AT commands.

We can check if the AT system works correctly with commands “AT”, we sent the “AT” command and we received the response “OK” that means the module and the AT system work correctly.

Commands	**Function**
AT+CWMODE=mode	Change Wifi Mode
AT+CWJAP=ssid,pwd	Connect to AP
AT+CWLAP	Lists available APs
AT+CIPSTART=type,addr,port	Establish TCP connection or register UDP port and start a connection
AT+CIPSEND	Send Data in tcp connection
AT+CIPCLOSE	Close TCP or UDP connection
AT+CIFSR	Get local IP address

Figure 2.18 Some AT commands

Next, we changed the default module mode to station mode (client) using command “AT+CWMODE”

Command	Response
AT+CWMODE=mode	OK

The mode is an integer value varies from 1 to 3 values, each value has a specific meaning:

1 = Station mode (client)

2 = AP mode (host)

3 = AP + Station mode (Yes, ESP8266 has a dual mode)

1.1.2.2 GPS module

Figure 2.19 GPS module GY-NEO6MV2

GPS GY-NEO6MV2 module flight controller with antenna for Arduino, it used to identify the robot geolocation position (latitude/longitude) which in turn sent via internet to the server and user systems, it plugged to serial communication (Tx/Rx) like the Wi-Fi module.

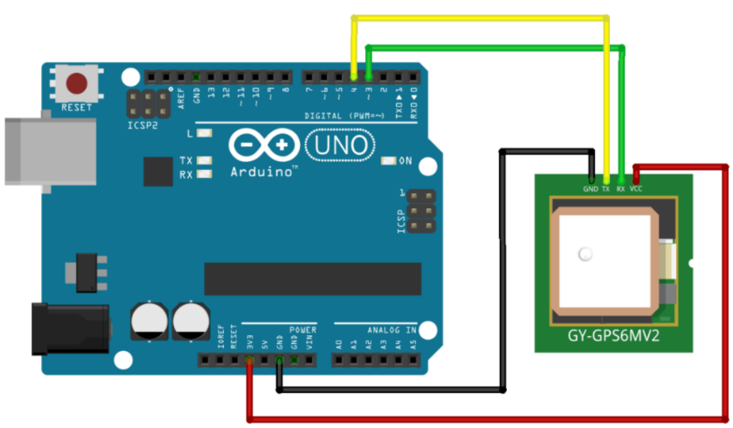

Figure 2.20 GPS module wiring

1.1.2.3 Ultrasonic module

Figure 2.21 Ultrasonic Module HC-SR04

The HC-SR04 ultrasonic sensor uses sonar to determine obstacles. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. From 2cm to 400 cm. Its operation is not affected by sunlight or black material like Sharp rangefinders are (although acoustically soft materials like cloth can be difficult to detect). It comes complete with ultrasonic transmitter and receiver module.

This module It emits an ultrasound at 40 000 Hz which travels through the air and if there is an object or obstacle on its path. Based on the time of flight and the speed of the sound we can calculate the distance.

For example, if the object is 10 cm away from the sensor, and the speed of the sound is 340 m/s or 0.034 cm/µs the sound wave will need to travel about 294 u seconds. But what we will get from the Echo pin will be double than number because the sound wave needs to travel forward and bounce backward. So, to get the distance in cm we need to multiply the received travel time value from the echo pin by 0.034 and divide it by 2.

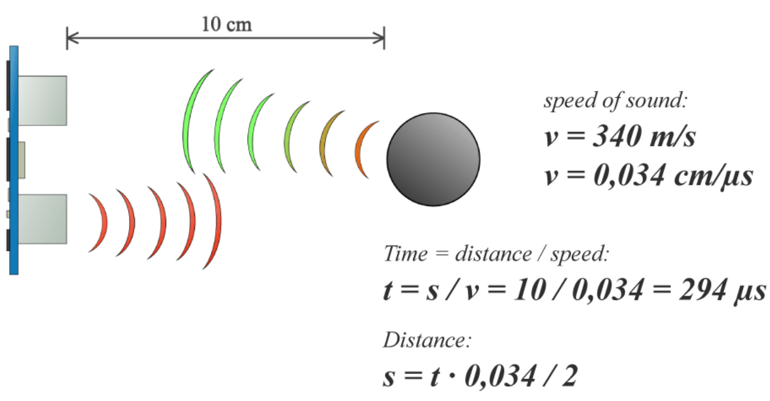

Figure 2.22 How ultrasonic works

We use this module to inform the user if robot faced an obstacle with the distance between robot and obstacle.

This module have four inputs: VCC input for the power , GND input for the ground, Trig and Echo to send information to the Arduino.

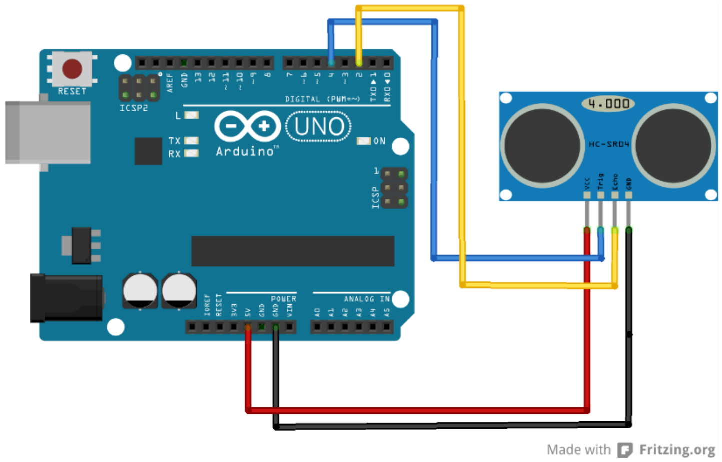

Figure 2.23 Ultrasonic module wiring

1.1.3 Motor driver shield

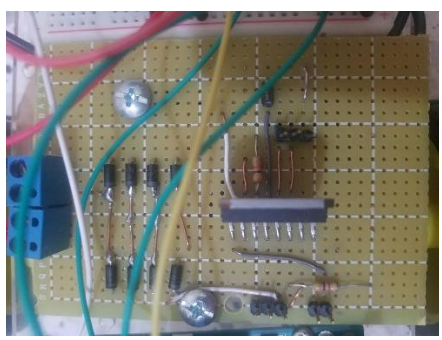

Figure 2.24 Motor driver shield

The technique used to move the robot to the right or left direction is to run in inverse the two side motors. The Arduino has not the capability to run motors except in one sense and cannot reverse motor direction, it means, the Arduino cannot send a negative value to motors, so we used motor driver shield based on IC L298 Following it datasheet. The result got a driver motor has 10 inputs:

2 inputs for the first motor (+/-)

2 inputs for the second motor (+/-)

2 digital inputs to send binary values to change the sense of first motor

3 digital inputs to send binary values to change the sense of second motor

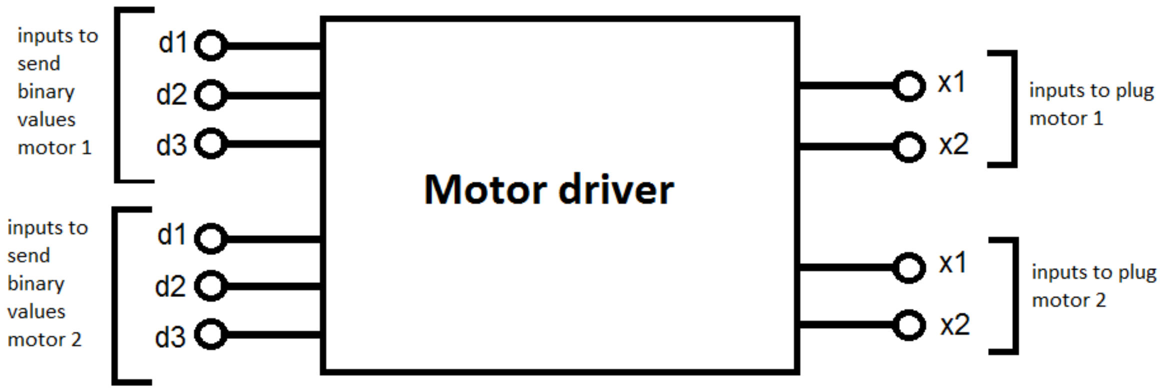

Figure 2.25 Motor driver architecture

The following table shows how we change the motor sense with binary values sent:

Entrées			Moteur
Enable A	Input 1	Input 2	
0	X	X	Roue libre
1	0	0	Stop
1	0	1	Sens 1
1	1	0	Sens 2
1	1	1	Stop

Figure 2.26 Motor Drive Truth table

According to the table we can control the robot sense by sending the appropriate values to get the desired direction.

1.1.4 Breadboard

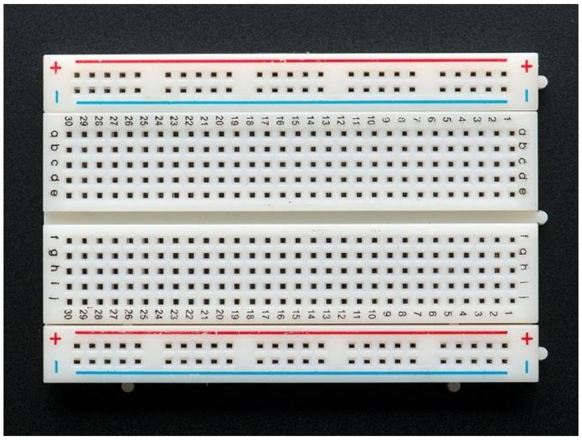

Figure 2.27 Breadboard

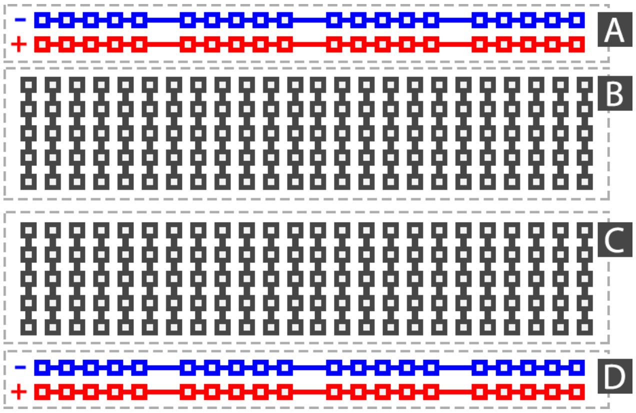

Figure 2.28 Breadboard architecture

The breadboard is referring to a solderless breadboard. These are great units for making temporary circuits and prototyping, and they require absolutely no soldering.

Another common use of breadboards is testing out new parts, such as Integrated Circuits (ICs). When we are trying to figure out how a part works and constantly rewiring things, we don't need to realize connection each time.

1.2 Diagrams

We using UML ,it is stands for Unified Modeling Language. It’s a rich language to model software solutions, application structures, system behavior and business processes.We choose Use Case diagram and Sequence diagram to represent our project.

1.2.1 Use Case diagram

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors).

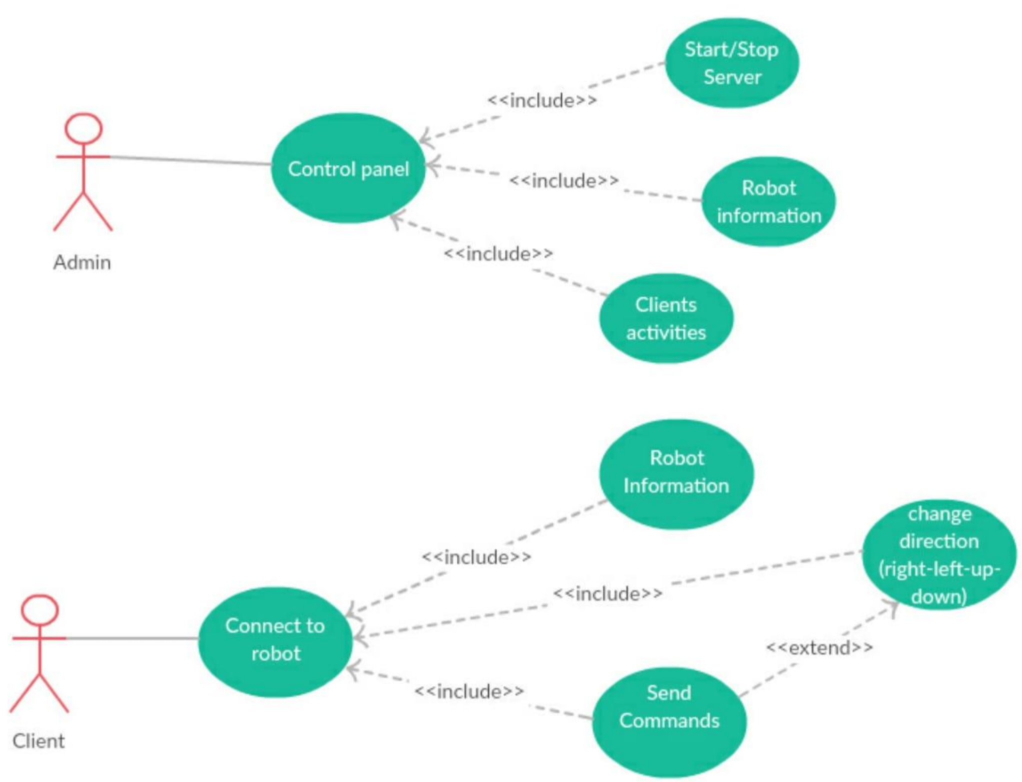

Figure 3.1 Use Case diagram

1.2.2 Sequence diagram

Sequence diagram is the most common kind of interaction diagram, which focuses on the message interchange between a number of lifelines.

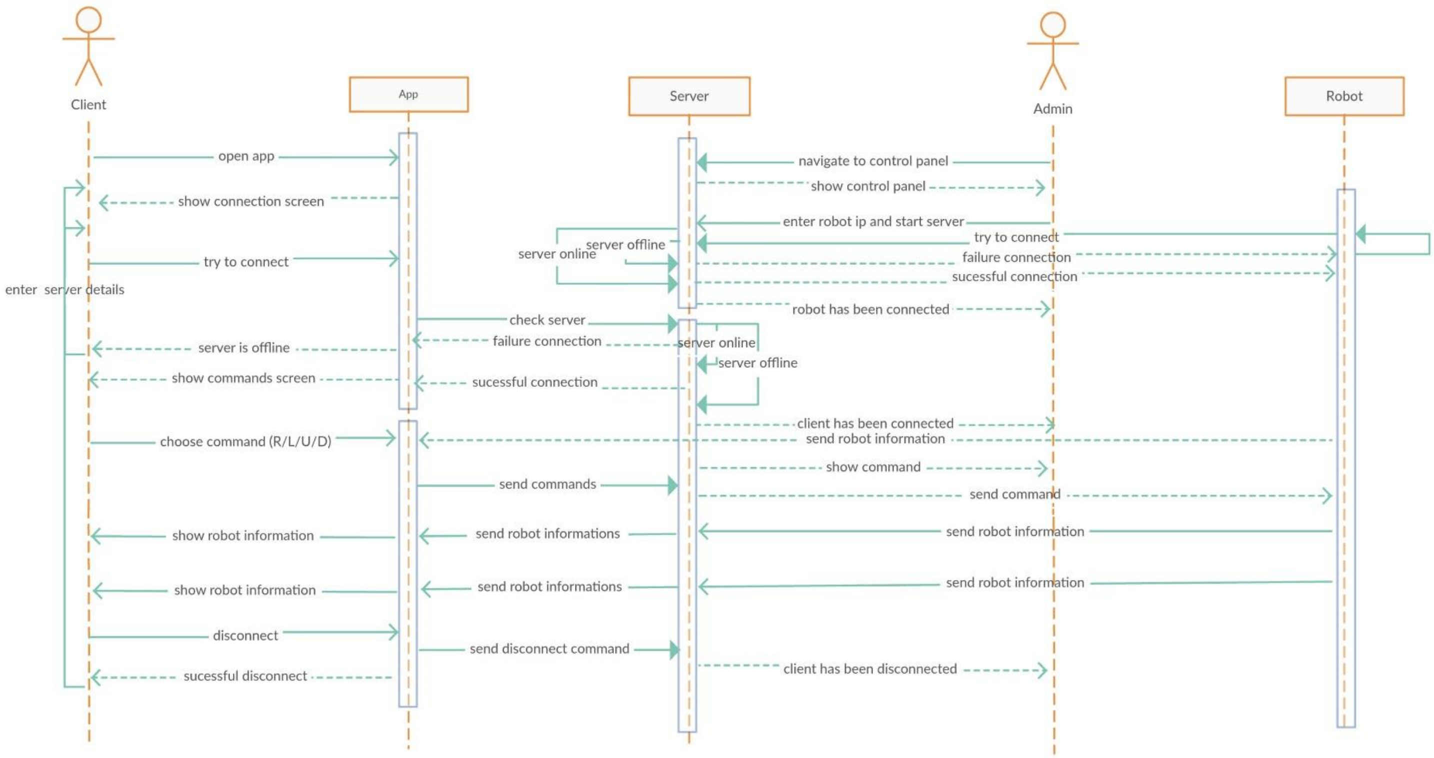

Figure 3.2 Sequence diagram

1.3 Software architecture

This section describes the software part responsible on the performing the interaction between the vehicle, the server and the mobile application. We, first, will show the system algorithm that controls the vehicle and implemented in the Arduino board. Then, we will talk about the architecture if the server, as well as, the android-based mobile application. Finally, we will talk about the commands system that ensures the communication and requests transmission between different project components.

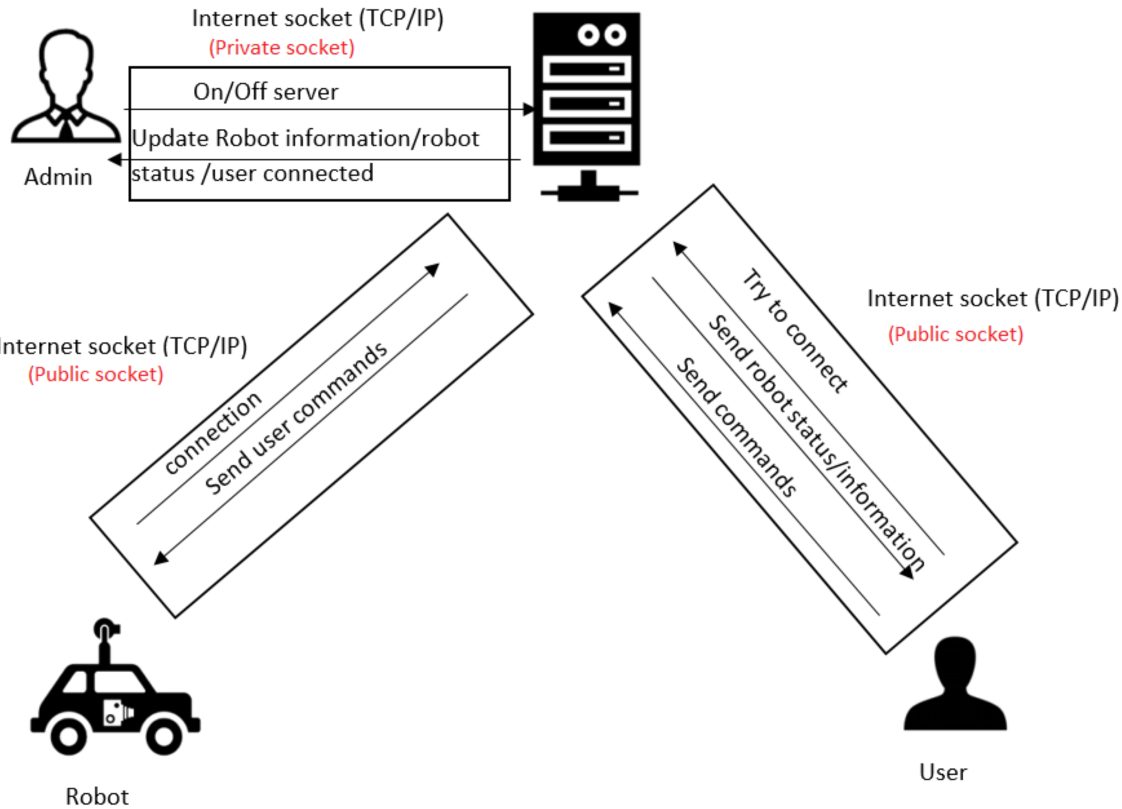

Figure 3.3 Communication Methodology

1.3.1 Server

The server system is a Node.js-based i.e. JavaScript runtime framework. Node.js allows for opening TCP/IP protocol communication sockets. There are two types of sockets, the localhost socket that is responsible on the connection and disconnection actions and the public socket that will be used for the internet-based requests.

The user interface of the service administration was built using the Node.js with Materialize theme.

When a connection is released between server and mobile application, an object socket of the user will be registered. If other users try to connect to the vehicle, they will be added to the waiting list and will notify the users that the vehicle is temporary unavailable.

After connection is released, the connected user will be able to control the vehicle remotely.

1.3.2 Mobile

The mobile application in destined to translate the human orders into commands and send them to the server. The application developed using Java language linked to the Android Studio IDE.

1.3.3 robot

The robot system developed using the C language and the Arduino IDE interface. The setup function role is to initialize the pins and the different modules. The robot can connect to the server system using its IP address and port number.

```
void setup() {

    init_motor_pins();
    init_ultrasonic();
    init_leds();
    serialWifi.begin(115200);
    //serialGPS.begin(4800);
    Serial.begin(115200);

    delay(500);

  // put your setup code here, to run once:
    boolean esp8266started = wifi.begin();
    delay(1000);
    Serial.println(esp8266started);
    serialWifi.println("AT+RST");
    delay(1000);
  if(!apConnected){
    Serial.println("Wifi connecting...");
    if(!apConnected)

while(!serverConnected){
  apConnected = wifi.connectToAP(_ssid, _pass);
   Serial.println("Connecting To Server...");
   serialWifi.println("AT+CIPSTART=\"TCP\",\"192.168.43.66\",6969");

    if(serialWifi.find("OK")){serverConnected=1;Serial.println("Server
Connected!");digitalWrite(led_red_Pin, HIGH);};
    delay(500);

    }
```

```
   }
}
```

The Loops function presents an infinity looping that, continuously, check the received requests from the Wi-Fi using the serial port. Simultaneously, the Arduino will use the GPS module to get real time coordinates and sends it to the server system.

```
  void loop(){
       //while(serialWifi.available()){

    String str= serialWifi.readString();

      Serial.println(str);


 if(str.indexOf("4001")>-1){
    digitalWrite(led_red_Pin, LOW);
  digitalWrite(led_green_Pin, HIGH);
  //up
  move_up();
  }
  else if (str.indexOf("4002")>-1){
      digitalWrite(led_red_Pin, LOW);
  digitalWrite(led_green_Pin, HIGH);
  //right
   move_right();
  }
  else if (str.indexOf("4003")>-1){
      digitalWrite(led_red_Pin, LOW);
  digitalWrite(led_green_Pin, HIGH);
  //left
   move_left();
  }
  else if (str.indexOf("4004")>-1){
      digitalWrite(led_red_Pin, LOW);
  digitalWrite(led_green_Pin, HIGH);
  //down
   move_down();
  }
  else if (str.indexOf("4000")>-1){
      digitalWrite(led_red_Pin, LOW);
  digitalWrite(led_green_Pin, HIGH);
  //stop
   stop_move();
  }
```

```
  else if (str.indexOf("3000")>-1){
  //use has connected
  digitalWrite(led_red_Pin, LOW);
  digitalWrite(led_green_Pin, HIGH);
  }
   else if (str.indexOf("404")>-1){
  //use has disconnected
  digitalWrite(led_red_Pin,HIGH );
  digitalWrite(led_green_Pin, LOW);
  }
  //}
 if(!serialWifi.available() && sendDataMetro.check()){

   send_data(get_obstacle_distance());
   delay(1000);
  }
    }
```

The commands sent to the robot in the form of numbers of 4 digits.

Code	**description**
4001	Move up
4002	Move right
4003	Move left
4004	Move down
4000	Stop
3000	Client has been connected
404	Client has been disconnected

Figure 3.4 Robot commands

Finally, we implemented the code that ensures the obstacles detection using the ultrasonic sensor to reach the user about the necessary information for safe navigation.

```
char *  get_obstacle_distance(){
                 // Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
```

```
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
long duration = pulseIn(echoPin, HIGH);
// Calculating the distance
int distance= duration*0.034/2;


char dst[10]= "dis_";
itoa(distance, dst+4, 10);
strcat(dst, "_tanc");
return dst;
}
```

2 Presentation

In this section we will present our project application

2.2 Admin interfaces

This interface contains a map and a button to on/off server and also a field to enter vehicle IP and also contains the information that sent by robot and the status of the vehicle and the users connected

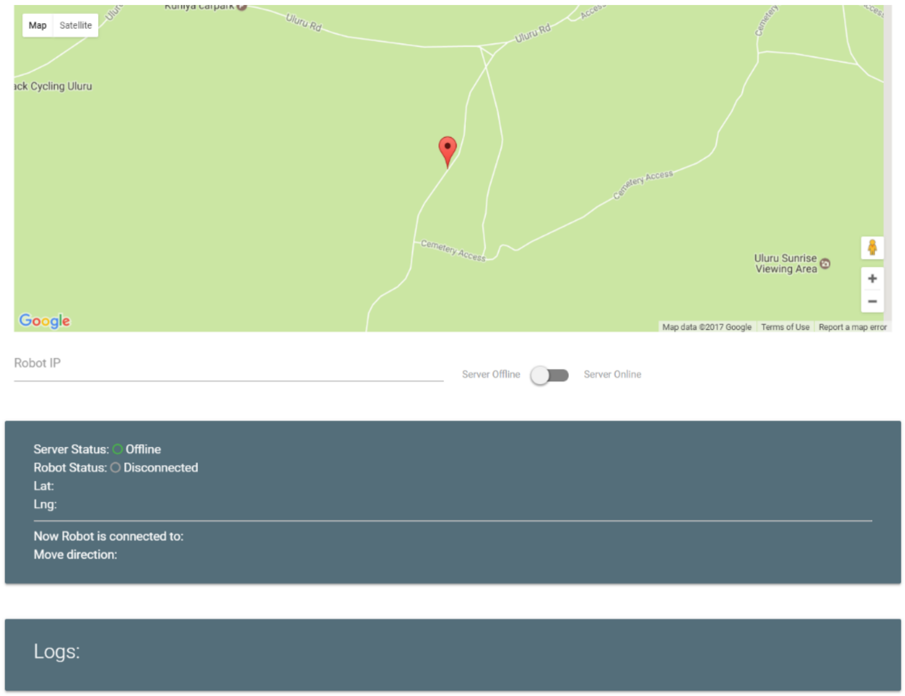

Figure 3.5 Admin UI

The below image shows when vehicle user connected to robot

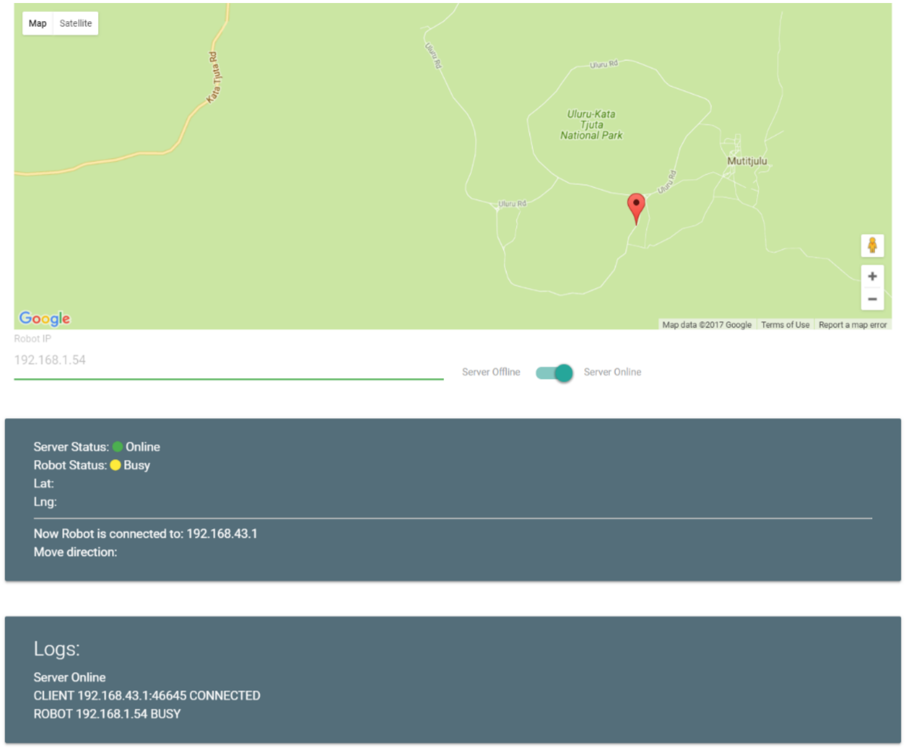

Figure 3.6 Admin UI when vehicle is busy

2.3 User interface

The first interface contains two fields to enter server IP address and port

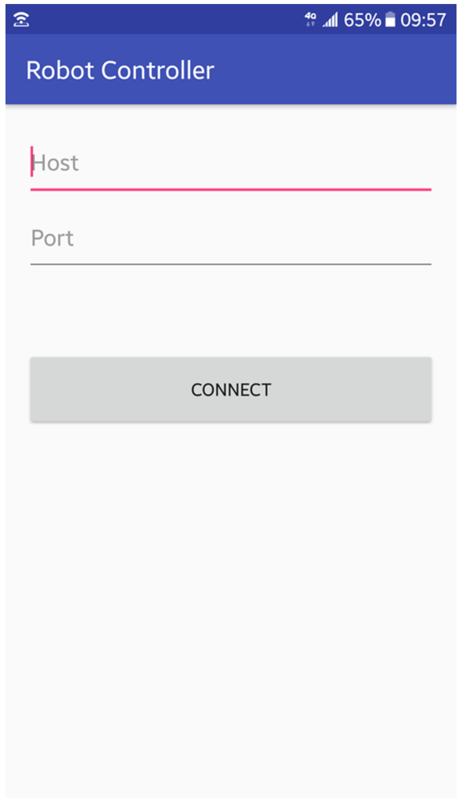

Figure 3.7 Interface mobile app to connect server

The Next interface contains a map and vehicle information, and the buttons to move the vehicle

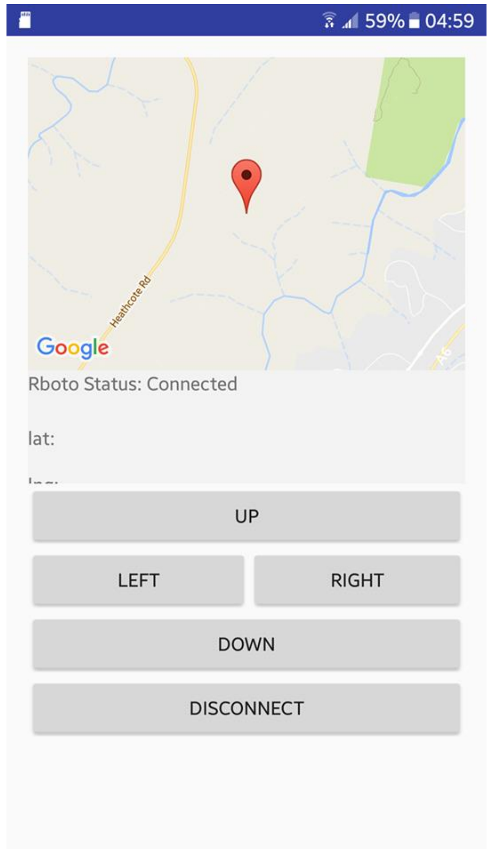

Figure 3.8 mobile app interface for vehicle commands

CONCLUSION

In this dissertation, we built a small ground vehicle that can be controlled remotely using Internet-based requests. We, sequentially, started by giving detailed information about the internet of things, and the Arduino hardware. Hence, this theoretic information will help to understand the general concepts related to the topic. Next, we expressed the practical part of this dissertation by presenting the vehicle and remote-operated commands architecture. The vehicle is, mainly, composed from the Arduino that has the capability to control and interact with different modules e.g. ultrasonic and motor driver shield modules. We implemented the Arduino system to ensure remote communication and receiving Internet-based requests, and the movement of the vehicle towards the four directions i.e. right, left, forward and back. Then we implemented the server system that ensures Internet-based requests transmission. Finally, we implemented an Android-based mobile application, which presents the interface layer responsible for interacting and translating the user requests and receive vehicle's information, and then we tested and examined the system with success. Based on the final results we demonstrated how newly available technologies can help governments that have not a high industrial power like the developed countries to reduce the importation of ready-made materials from foreign countries, which cost heavy losses.

FUTURE WORK

In the future work, we will try to improve and enhance this system by adding to new significant features. We will include to the system, the vision-based Simultaneous Localization and Mapping technology that helps the system to autonomously explore and navigate in unknown environments with the lack of external positioning technology e.g. GPS.

BIBLIOGRAPHY

[1] The internet of things : an overview - understanding the issues and challenges of a more connected world – Internet Society

[2] Simply Arduino - Abdullah Ali Abdullah

WEBSITES

[5] http://www.happiestminds.com/Insights/internet-of-things/

[6] http://www.dataversity.net/brief-history-internet-things/

[7] http://www.businessinsider.com/internet-of-things-in-healthcare-2016-8

[8] https://www.itic.org/policy/internet-of-things

[9] https://redshift.autodesk.com/industrial-internet-of-things-iot-terms/

[10] http://www.businessinsider.com/internet-of-things-smart-agriculture-2016-10

[11]http://www.wireless-mag.com/Features/40269/how-the-internet-of-things-is-impacting-public-safety-.aspx

[12]http://www.thewhir.com/web-hosting-news/the-four-internet-of-things-connectivity-models-explained

[13] https://www.linkedin.com/pulse/societal-impact-internet-things-bharat-joshi

[14] http://droneanalyst.com/2014/12/01/drones-are-the-future-of-iot/

[15] http://www.businessinsider.com/wearable-technology-iot-devices-2016-8

[16] https://readwrite.com/2016/12/29/8-predictions-for-the-future-of-iot-in-2017-il1/

[17] https://www.arduino.cc/en/guide/introduction

[18] http://creativityprojects.blogspot.com/2013/03/history-of-arduino_4195.html

[19] https://www.tutorialspoint.com/arduino/arduino_board_description.htm

[20]http://www.ebay.com/gds/8-Types-of-Arduino-Boards-to-Choose-From-/10000000178608458/g.html

ملخص

مع وجود الإنترنت ودورها الهام في حياتنا، يسعى الجميع لاستخدامه في جميع جوانب الحياة لتوفير الوقت والمال. كانت هناك أنواع كثيرة من البحوث حول ربط أي شيء المحيط بنا أو النظام البيئي بالإنترنت. أيضا، هناك إمكانية لأشياء التحكم عن بعد. وقد جاء إنترنت الأشياء بسبب هذه الاحتياجات. وتستند هذه التكنولوجيا على ربط كل شيء بالإنترنت والتحكم عن بعد إما عن طريق تطبيقات الهاتف، والمواقع أو حتى من خلال الخدمات السحابية.

والهدف من هذه الأطروحة هو بناء مركبة صغيرة تعمل عن بعد. ويعرض دراسة تجريبية في كيفية بناء الإنترنت كاملة من الأشياء نظام باستخدام رخيصة ومتكررة ومتاحة. ويتكون المشروع من ثلاثة أجزاء أساسية، المركبة التي تتألف من لوحة الدوائر الكهربائية من أردوينو التي تتفاعل مع الوحدات النمطية، نظام اتصال بالانترنت عن طريق الويفي ونظام تحديد المواقع وحدات، ونظام الخادم الذي يضمن نقل الطلبات من المستخدم إلى مركبة ونقل المعلومات منها إلى المستخدم، وتطبيقات الهاتف المتحرك القائم على الروبوت الذي يعرض واجهة للتفاعل مع المستخدم.

وستشجع النتائج التي تم الحصول عليها تعزيز بعض أنواع الصناعات في الجزائر، على سبيل المثال. الأمن، واستكشاف والسيارات المتنقلة ذاتية القيادة. كما أنها سوف تثبت كيف من مكونات رخيصة يمكننا بناء منتجات مبتكرة حل العديد من المشاكل اليومية والقضاء على ضرورة استيراد المواد الجاهزة من الدول الأجنبية، مما يكلف الحكومة خسائر فادحة.

الكلمات المفتاحية: اردوينو -ربوت - تحكم عن بعد - انترنت الأشياء

Abstract

With the presence of the Internet and its significant role in our life, everyone seeks to use it in all aspects of life to save time and money. There have been many types of research about linking anything surrounding us or the ecosystem to Internet. Also, there is the possibility to remote control things. The Internet of things has come because of these needs. This technology is based on connecting everything to internet and remotely control them either by phone applications, websites or even through cloud services.

The aim of this dissertation is to build a remote-operated micro grounded vehicle. It presents an experimental study in how to build complete internet of things system using cheap and frequent available component. The project is composed from three essential parts i.e. the vehicle that composed from the Arduino circuit board that interacts with modules e.g. WIFI and GPS modules, the server system that ensures the requests transmission from the user to the grounded vehicle and the information transmission from the vehicle to the user, and the Android-based mobile application that presents the interface for interaction with the user.

The obtained results will encourage and strengthen some type of industries in Algeria e.g. security, exploring and self-driving mobile vehicles. As it will demonstrate how from cheap components we could build innovative products that solve many daily problems and eliminate the necessity to import ready-made materials from foreign countries, which cost the government big amount of transferred fund.

KEYWORD: Arduino - Robot - Remote Control - Internet of things

Résumé

Avec l'Internet et son important rôle dans notre vie, tout le monde cherche à l'utiliser dans tous les aspects de la vie, pour gagner du temps et de l'argent. Il y avait de nombreuses recherches sur l'association tout ce que nous entoure ou l'écosystème à l'Internet. En plus, il existe la possibilité de télécommander les objets que nous entourent. L'Internet des objets est née comme une suite logique à ces besoins. Cette technologie est basée sur la liaison à l'Internet et le contrôle à distance soit par des applications de téléphone, sites Web ou même grâce à des services de cloud.

L'objectif de cette thèse est de construire un petit véhicule télécommandé. Il présente une étude expérimentale sur la façon de construire un système complet d'internet des objets en utilisant composants accessibles financièrement et disponibles. Le projet est constitué en trois parties de base, le véhicule comprenant des circuits électriques de la plaque de Arduino qui interagissent avec les modules, et le système connexion Internet et le GPS, et le système de serveur qui assure le transfert des demandes de l'utilisateur au véhicule , l'application mobile sur la base du robot, qui affiche l'interface pour interagir avec l'utilisateur.

Les résultats obtenus vont encourager et favoriser certains types d'industries en Algérie, par exemple. La sécurité, l'exploration et les voitures auto-conduite. Ils permettront également de montrer comment à travers des composants pas cher, nous pouvons créer des produits innovants qui permettent de résoudre de nombreux problèmes quotidiens et d'éliminer la nécessité d'importer des matériaux des pays étrangers, ce qui peut coûter le gouvernement des sommes colossales.

MOT-CLE : Arduino - Robot - Contrôle à distance – Internet des objets